TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/H Series

TMP95C063FG
TMP95C063DFG

TOSHIBA CORPORATION

Semiconductor Company

Preface

Thank you very much for making use of Toshiba microcomputer LSIs.
Before use this LSI, refer the section, "Points of Note and Restrictions".
Especially, take care below cautions.

CAUTION

How to release the HALT mode

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, $\overline{\text{NMI2}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE or STOP mode. (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

Document Change Notification

The purpose of this notification is to inform customers about the launch of the Pb-free version of the device. The introduction of a Pb-free replacement affects the datasheet. Please understand that this notification is intended as a substitute for a revision of the datasheet.

Changes to the datasheet may include the following, though not all of them may apply to this particular device.

1. Part number

   Example: TMPxxxxxxF → TMPxxxxxxFG

   All references to the previous part number were left unchanged in body text. The new part number is indicated on the prelims pages (cover page and this notification).

2. Package code and package dimensions

   Example: LQFP100-P-1414-0.50C → LQFP100-P-1414-0.50F

   All references to the previous package code and package dimensions were left unchanged in body text. The new ones are indicated on the prelims pages.

3. Addition of notes on lead solderability

   Now that the device is Pb-free, notes on lead solderability have been added.

4. RESTRICTIONS ON PRODUCT USE

   The previous (obsolete) provision might be left unchanged on page 1 of body text. A new replacement is included on the next page.

5. Publication date of the datasheet

   The publication date at the lower right corner of the prelims pages applies to the new device.

1. Part number

2. Package code and dimensions

Previous Part Number (in Body Text)	Previous Package Code (in Body Text)	New Part Number	New Package Code
TMP95C063F	P-QFP144-2020-0.50	TMP95C063FG	QFP144-P-2020-0.50
_ (Note)	_ (Note)	TMP95C063DFG	LQFP144-P-2020-0.50D

Note: Pb-containing variant not available.

*: For the dimensions of the new package, see the attached Package Dimensions diagram.

3. Addition of notes on lead solderability

The following solderability test is conducted on the new device.

Solderability of lead free products

Test Parameter	Test Condition	Note
Solderability	Use of Sn-37Pb solder Bath Solder bath temperature = 230°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: Solderability rate until forming ≥ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux (use of lead free)	

4. RESTRICTIONS ON PRODUCT USE

The following replaces the "RESTRICTIONS ON PRODUCT USE" on page 1 of body text.

RESTRICTIONS ON PRODUCT USE 20070701-EN

- The information contained herein is subject to change without notice.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc.
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in his document shall be made at the customer's own risk.
- The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties.
- Please contact your sales representative for product-by-product details in this document regarding RoHS compatibility. Please use these products in this document in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances. Toshiba assumes no liability for damage or losses occurring as a result of noncompliance with applicable laws and regulations.
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions.

5. Publication date of the datasheet

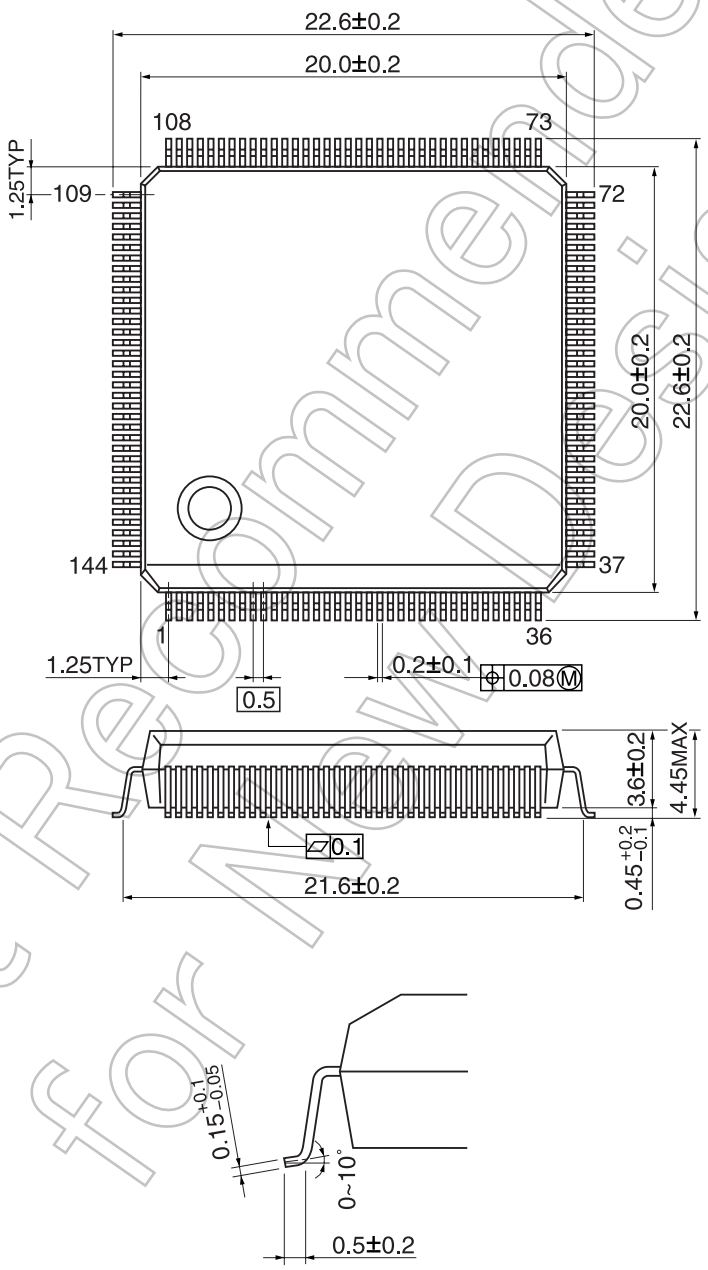
The publication date of this datasheet is printed at the lower right corner of this notification.

(Annex)

Package Dimensions

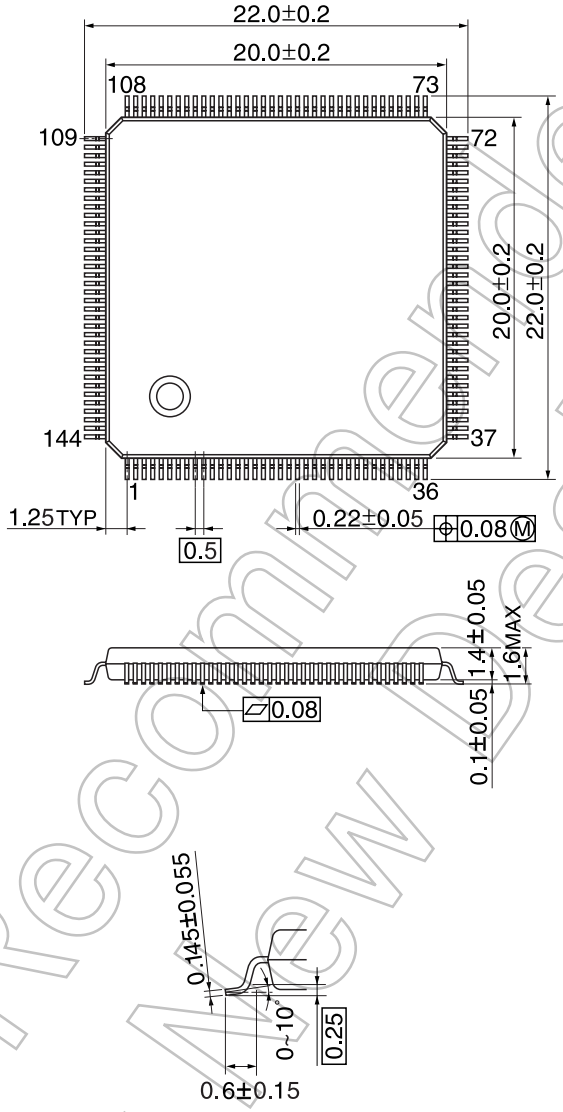
QFP144-P-2020-0.50

Unit: mm



LQFP144-P-2020-0.50D

Unit: mm



CMOS 16-Bit Microcontroller

TMP95C063F

1. Outline and Features

TMP95CO63F was developed as a high-speed, advanced 16-bit microcontroller for a range of mid- to large-scale equipment.

TMP95CO63F is presented in a 144-pin plastic flat package. Its features are as follows.

(1) Original high-speed 16-bit CPU (900H_CPU)
- Instruction mnemonics upwardly compatible with TLCS-90/900
- 16M-byte linear address space
- General-purpose registers using register bank system
- 16-bit multiplication / division instructions, bit transfer / arithmetic instructions
- Micro DMA: four channels (640 ns / 2 bytes at 25 MHz)

(2) Minimum instruction execution time: 160 ns (at 25 MHz)

(3) Internal RAM : No
Internal ROM : No

(4) External memory expansion
- Expandable to 16 Mbytes (common to programs and data)
- External data bus width selection pin (AM8 / $\overline{16}$)
- Can use both 8- and 16-bit external buses
...dynamic data bus sizing

(5) Internal DRAM controller: two channels
- 2$\overline{CAS}$ / 2$\overline{WE}$ selectable

(6) 8-bit timer : eight channels

(7) 16-bit timer : two channels

(8) Pattern generator : four bits, two channels

(9) General-purpose serial interface : two channels
- Baud rate generated by external clock

(10) 10-bit AD converter : eight channels

(11) 8-bit DA converter : two channels

(12) Watchdog timer

(13) Chip selector, wait controller : four blocks

(14) Interrupt function
- CPU interrupts : 2 (software interrupt instructions, illegal instructions)
- Internal interrupts: 22 (seven priority levels available)
- External interrupts: 11 (seven priority levels available)

(15) Input / output ports
91 pins

(16) Standby function
Three HALT modes (RUN, IDLE, STOP)

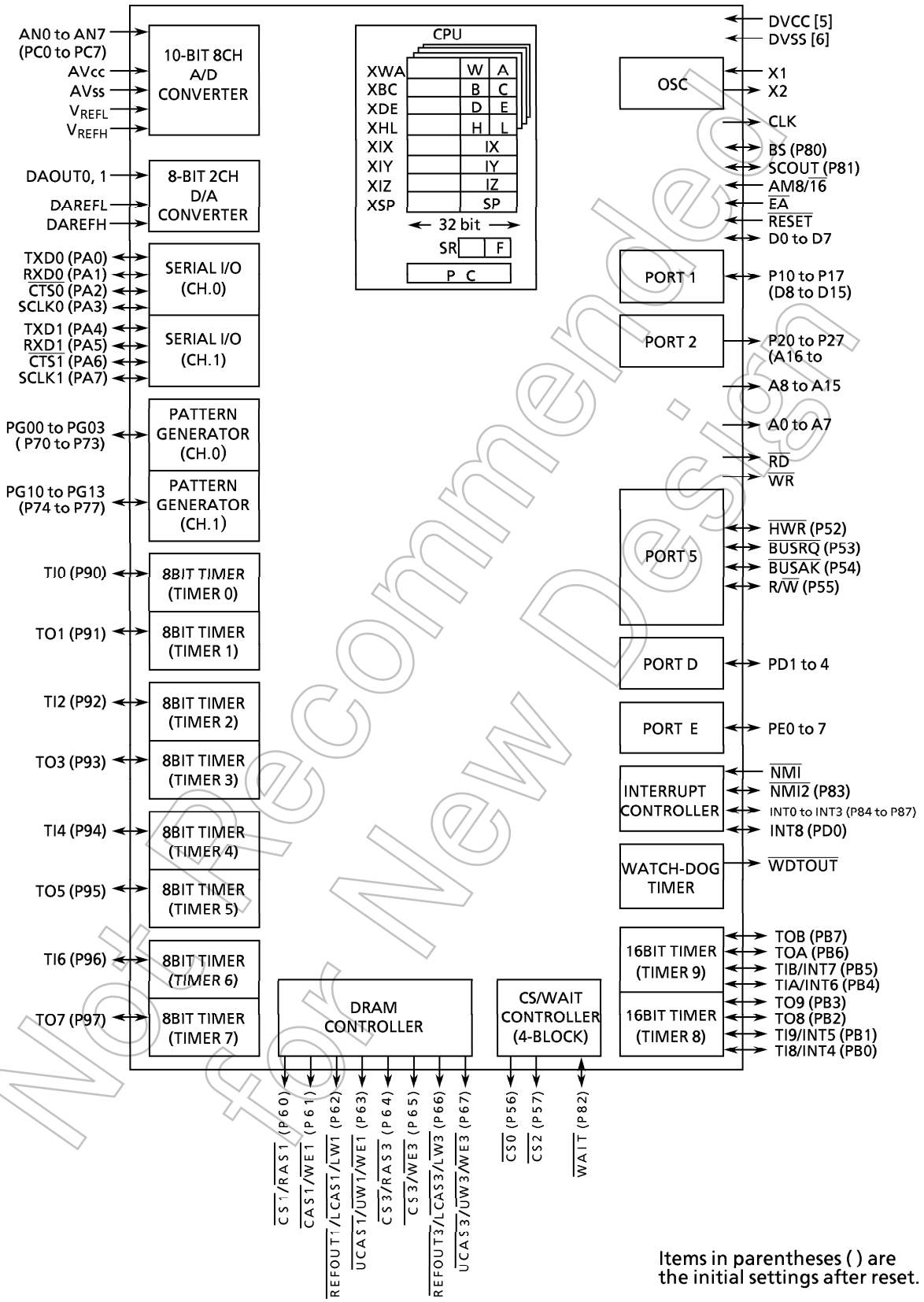


Figure 1 TMP95CO63 Block Diagram

2. Pin Assignment and Functions

2.1 Pin Assignment (Top view)

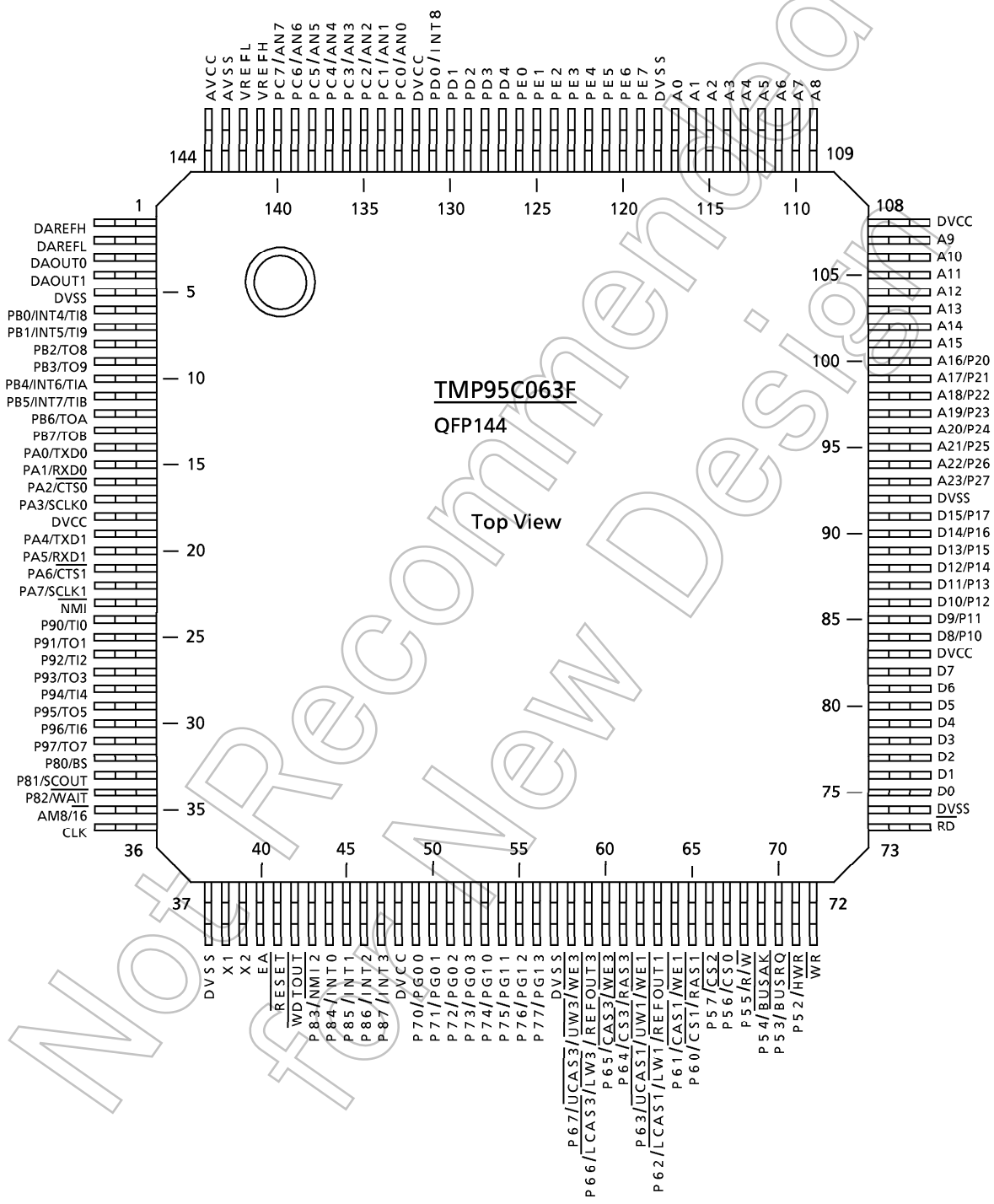


Figure 2.1 Pin Assignment

2.2 Pin Names and Functions

Table 2.2 shows the I/O pin names and their functions.

Table 2.2

Pin Name	Pin Number	Input / Output	Function
D0 to D7	8	Input / Output	Data : Data bus 0 to 7
P10 to P17 D8 to D15	8	Input / Output Input / Output	Port 1 : I/O ports. Individual pins can be set as inputs or outputs. Data : Data bus 8 to 15
P20 to P27 A16 to A23	8	Output Output	Port 2 : Output-only ports Address : Address bus 16 to 23
A8 to A15	8	Output	Address : Address bus 8 to 15
A0 to A7	8	Output	Address : Address bus 0 to 7
$\overline{RD}$	1	Output	Read : Strobe signal to read external memory (Setting bit 0 of the P5 register (RDE) to "0" outputs $\overline{RD}$ even when reading internal areas.)
$\overline{WR}$	1	Output	Write : Strobe signal to write data of pins D0 to 7.
P52 $\overline{HWR}$	1	Input / Output Output	Port 52 : I/O port (with pull-up) Upper write: Strobe signal for writing data of pins D8 to 15.
P53 $\overline{BUSRQ}$	1	Input / Output Input	Port 53 : I/O port (with pull-up) Input Bus request: Signal to request following pins set to high impedance: D0 to 15, A0 to 23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{RAS}$, $\overline{CAS}$, and $\overline{REFOUT}$ (*). (for external DMAC)
P54 $\overline{BUSAK}$	1	Input / Output Output	Port 54 : I/O port (with pull-up) Input Bus acknowledge: Signal to indicate following pins set to high impedance in response to $\overline{BUSRQ}$ signal: D0 to 15, A0 to 23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{RAS}$, $\overline{CAS}$, and $\overline{REFOUT}$ (*). (for external DMAC)
P55 R/$\overline{W}$	1	Input / Output Output	Port 55 : I/O port (with pull-up) Read/write: "1" indicates read or dummy cycle; "0" indicates write cycle.
P56 $\overline{CS0}$	1	Output Output	Port 56 : Output-only port Chip select 0: Outputs 0 if address is within specified address range.
P57 $\overline{CS2}$	1	Output Output	Port 57 : Output-only port Chip select 2: Outputs 0 if address is within specified address range.

Note : The external DMA controller, which uses the $\overline{BUSRQ}$ and $\overline{BUSAK}$ pins, cannot access the internal memory or internal I/O of TMP95C063.

(*) The DRAM control pins are high impedance only when bus release mode is set by the DRAM controller. For details, see 3.7, Dynamic RAM (DRAM) Controller.

Pin name	Pin Number	Input / Output	Function
P60 $\overline{CS1}$ $\overline{RAS1}$	1	Output Output Output	Port 60: Output-only port Chip select 1: Outputs 0 if address is within specified address range. Low address strobe 1: Outputs $\overline{RAS}$ strobe for DRAM if address is within specified address range.
P61 $\overline{CAS1}$ $\overline{WE1}$	1	Output Output Output	Port 61: Output-only port Column address strobe 1: Outputs 0 if address is within specified address range. (8-bit bus or 2WE mode) Write enable 1: Outputs write enable signal for DRAM. (2CAS mode)
P62 $\overline{LCAS1}$ $\overline{LW1}$ $\overline{REFOUT1}$	1	Output Output Output Output	Port 62: Output-only port Lower column address strobe 1: Outputs lower CAS strobe for DRAM if address is within specified address range. (2CAS mode) Lower write enable 1: Outputs lower write enable signal for DRAM. (2WE mode) Refresh out 1: 0 indicates generation of refresh cycle. (8-bit bus mode)
P63 $\overline{UCAS1}$ $\overline{UW1}$ $\overline{WE1}$	1	Output Output Output Output	Port 63: Output-only port Upper column address strobe 1: Outputs upper CAS strobe for DRAM if address is within specified address range. (2CAS mode) Upper write enable 1: Outputs upper write enable signal for DRAM. (2WE mode) Write enable 1: Outputs write enable signal for DRAM. (8-bit bus mode)
P64 $\overline{CS3}$ $\overline{RAS3}$	1	Output Output Output	Port 64: Output-only port Chip select 3: Outputs 0 if address is within specified address range. Low address strobe 3: Outputs $\overline{RAS}$ strobe for DRAM if address is within specified address range.
P65 $\overline{CAS3}$ $\overline{WE3}$	1	Output Output Output	Port 65: Output-only port Column address strobe 3: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address range. (8-bit bus or 2WE mode) Write enable 3: Outputs write enable signal for DRAM. (2CAS mode)
P66 $\overline{LCAS3}$ $\overline{LW3}$ $\overline{REFOUT3}$	1	Output Output Output Output	Port 66: Output-only port Lower column address strobe 3: Outputs lower CAS strobe for DRAM if address is within specified address range. (2CAS mode) Lower write enable 3: Outputs lower write enable signal for DRAM. (2WE mode) Refresh out 3: 0 indicates generation of refresh cycle. (8-bit bus mode)

Pin name	Pin Number	Input / Output	Function
P67 $\overline{UCAS3}$ $\overline{UW3}$ $\overline{WE3}$	1	Output Output Output Output	Port 67: Output-only port Upper column address strobe 3: Outputs upper CAS strobe for DRAM if address is within specified address range. (2CAS mode) Upper write enable 3: Outputs upper write enable signal for DRAM. (2WE mode) Write enable 3: Outputs write enable signal for DRAM. (8-bit bus mode)
P70 to P73 PG00 to PG03	4	Input / Output Output	Ports 70 to 73: I/O ports. Individual pins can be set as inputs or outputs. (with pull-up) Pattern generator ports 00-03
P74 to P77 PG10 to PG13	4	Input / Output Output	Ports 74 to 77: I/O ports. Individual pins can be set as inputs or outputs. (with pull-up) Pattern generator ports 10-13
P80 BS	1	Input / Output Output	Port 80: I/O port (with pull-up) Bus start: Indicates start of bus cycle.
P81 SCOUT	1	Input / Output Output	Port 81: I/O port (with pull-up) System clock output: Outputs system clock (external clock divided by 2).
P82 $\overline{WAIT}$	1	Input / Output Input	Port 82: I/O port (with pull-up) Wait: CPU bus wait request pin. (1 + N or 0 + NWAIT mode)
P83 $\overline{NMI2}$	1	Input / Output Input	Port 83: I/O port (with pull-up) Non-maskable interrupt request pin 2: Falling-edge interrupt request pin
P84 INT0	1	Input / Output Input	Port 84: I/O port (with pull-up) Interrupt request pin 0: Can be programmed for level or rising-edge detection.
P85 INT1	1	Input / Output Input	Port 85: I/O port (with pull-up) Interrupt request pin 1: Rising-edge interrupt request pin
P86 INT2	1	Input / Output Input	Port 86: I/O port (with pull-up) Interrupt request pin 2: Rising-edge interrupt request pin
P87 INT3	1	Input / Output Input	Port 87: I/O port (with pull-up) Interrupt request pin 3: Rising-edge interrupt request pin
P90 TI0	1	Input / Output Input	Port 90: I/O port (with pull-up) Timer input 0: Timer 0 input
P91 TO1	1	Input / Output Output	Port 91: I/O port (with pull-up) Timer output 1: Timer 0 or 1 output
P92 TI2	1	Input / Output Input	Port 92: I/O port (with pull-up) Timer input 2: Timer 2 input
P93 TO3	1	Input / Output Output	Port 93: I/O port (with pull-up) Timer output 3: Timer 2 or 3 output
P94 TI4	1	Input / Output Input	Port 94: I/O port (with pull-up) Timer input 4: Timer 4 input

Pin name	Pin Number	Input / Output	Function
P95 TO5	1	Input / Output Output	Port 95: I/O port (with pull-up) Timer output 5: Timer 4 or 5 output
P96 TI6	1	Input / Output Input	Port 96: I/O port (with pull-up) Timer input 6: Timer 6 input
P97 TO7	1	Input / Output Output	Port 97: I/O port (with pull-up) Timer output 7: Timer 6 or 7 output
PA0 TXD0	1	Input / Output Output	Port A0: I/O port (with pull-up) Serial transmit data output 0
PA1 RXD0	1	Input / Output Input	Port A1: I/O port (with pull-up) Serial receive data input 0
PA2 $\overline{\text{CTS0}}$	1	Input / Output Input	Port A2: I/O port (with pull-up) Serial data clear to send 0
PA3 SCLK0	1	Input / Output Input / Output	Port A3: I/O port (with pull-up) Serial clock input/output 0
PA4 TXD1	1	Input / Output Output	Port A4: I/O port (with pull-up) Serial data output 1
PA5 RXD1	1	Input / Output Input	Port A5: I/O port (with pull-up) Serial data input 1
PA6 $\overline{\text{CTS1}}$	1	Input / Output Input	Port A6: I/O port (with pull-up) Serial data clear to send 1
PA7 SCLK1	1	Input / Output Input / Output	Port A7: I/O port (with pull-up) Serial clock input / output 1
PB0 TI8 INT4	1	Input / Output Input Input	Port B0: I/O port (with pull-up) Timer input 8: Used as count or capture trigger input for timer 8. Interrupt request pin 4: Can be programmed for rising- or falling-edge detection.
PB1 TI9 INT5	1	Input / Output Input Input	Port B1: I/O port (with pull-up) Timer input 9: Used as count or capture trigger input for timer 8. Interrupt request pin 5: Rising-edge interrupt request pin
PB2 TO8	1	Input / Output Output	Port B2: I/O port (with pull-up) Timer output 8: Timer 8 output pin
PB3 TO9	1	Input / Output Output	Port B3: I/O port (with pull-up) Timer output 9: Timer 8 output pin
PB4 TIA INT6	1	Input / Output Input Input	Port B4: I/O port (with pull-up) Timer input A: Used as count or capture trigger input for timer 9. Interrupt request pin 6: Can be programmed for rising or falling edge detection.
PB5 TIB INT7	1	Input / Output Input Input	Port B5: I/O port (with pull-up) Timer input B: Used as count or capture trigger input for timer 9. Interrupt request pin 7: Rising-edge interrupt request pin

Pin Name	Pin Number	Input / Output	Function
PB6 TOA	1	Input / Output Output	Port B6: I/O port (with pull-up) Timer output A: Timer 9 output pin
PB7 TOB	1	Input / Output Output	Port B7: I/O port (with pull-up) Timer output B: Timer 9 output pin
PC0 to PC7 AN0 to AN7	8	Input Input	Input Port C: Input ports Analog inputs: A/D converter inputs
PD0	1	Input / Output Input	Port D0: I/O port (with pull-up) Interrupt request pin 8: Rising-edge interrupt request pin
PD1 to 4	4	Input / Output	Port D1 to D4: I/O ports (with pull-up)
PE0 to 7	8	Input / Output	Port E0 to E7: I/O ports (with pull-up)
DAREFH	1	Input	Reference voltage input pin for D/A converter (H)
DAREFL	1	Input	Reference voltage input pin for D/A converter (L)
DAOUT0	1	Output	D/A output 0: D/A converter 0 analog current output pin
DAOUT1	1	Output	D/A output 1: D/A converter 1 analog current output pin
$\overline{\text{WDTOUT}}$	1	Output	Watchdog timer output pin
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Falling-edge interrupt request pin. Can also be programmed as rising-edge interrupt request pin.
CLK	1	Output	Clock output: Outputs external input clock X1 divided by 4. Pulled up during reset.
$\overline{\text{EA}}$	1	Input	Fixed to ground.
AM8/$\overline{16}$	1	Input	Address mode: External data bus width selection pin. Set to 0 when using fixed 16-bit external bus or dual 8/16-bit external bus. Set to 1 with 8-bit external bus fixed.
$\overline{\text{RESET}}$	1	Input	Reset: Initializes LSI. (with pull-up)
VREFH	1	Input	Reference voltage input pin for A/D converter (H)
VREFL	1	Input	Reference voltage input pin for A/D converter (L)
AVCC	1		A/D converter power supply pin
AVSS	1		A/D converter ground pin (0 V)
X1/X2	2	Input / Output	Oscillator connecting pins
DVCC	5		Power supply pin (+ 5 V)
DVSS	6		Ground pin (0 V)

Note 1 : Apart from the RESET pin, the pull-up resistors can be disconnected by software.
Note 2 : Connect all DVCC and AVCC pins to power supply and all DVSS and AVSS pins to GND.

3. Operation

The following is a block-by-block description of the functions and basic operation of TMP95C063.

Note that the description concludes with cautions and restrictions for each block in 7, Usage Cautions and Restrictions.

3.1 CPU

TMP95C063 contains an advanced, high-speed 16-bit CPU (the 900H_CPU). The CPU is described in the TLCS-900 CPU section in the previous chapter.

The following describes the CPU functions unique to TMP95C063 that are not described in "TLCS-900 CPU".

3.1.1 Reset Operation

At TMP95C063 reset, the power supply voltage must be within the operating range and internal oscillation must be stable. Set the $\overline{\text{RESET}}$ input to 0 for at least ten system clocks (= 10 states: 0.8 μs for a 25-MHz clock).

When the reset is accepted, the CPU:

- Sets the program counter (PC) to the reset vector stored at addresses FFFF00H to FFFF02H.
  PC (7 : 0) ← value at address FFFF00H
  PC (15 : 8) ← value at address FFFF01H
  PC (23 : 16) ← value at address FFFF02H
- Sets the stack pointer (XSP) to 100H
- Sets bits IFF2 to 0 of the status register (SR) to 111 (this sets the interrupt level mask register to level 7).
- Sets the MAX bit of the status register (SR) to 1 (this sets maximum mode). (Note: This product does not support minimum mode. Do not use the MIN instruction.)
- Clears bits RFP2 to 0 of the status register (SR) to 000 (this sets the register banks to 0).

After reset is released, the CPU begins execution from the instruction at the location specified in the PC. Other than the changes described above, reset does not alter any internal CPU registers.

When reset is accepted, processing of the internal I/O, port, and other pins are as follows:

- Initializes the internal I/O registers as per specifications.
- Sets port pins (including pins also used as internal I/O) to general-purpose input or output mode.
- Sets the $\overline{\text{WDTOUT}}$ pin to 0 (watchdog timer is enabled after reset).
- Pulls up the clock pin to 1.

3.1.2 External Data Bus Width Selection Pin (AM8/$\overline{16}$)

After reset,TMP95C063 automatically operates in either 8 or 16-bit bus mode depending on the AM8/$\overline{16}$ pin setting.

- For dual 8- and 16-bit external data bus, or fixed 16-bit external data bus operation
  Set the AM8/16 pin to 0. Fixes port 1 (P1) to D8-15.
  Note that the external data bus width is set by the chip select / wait control register described in 3.6.1.

- For fixed 8-bit external data bus operation
  Set the AM8/$\overline{16}$ pin to 1. Sets port 1 (P1) to port mode.
  The chip select / wait control register values described in 3.6.1 (<B0BUS>, <B1BUS>, <B2BUS>, <B3BUS>, and <BEXBUS>) are ignored and the device can only operate with an 8-bit external data bus.
  However, when using the DRAM controller, it is necessary to set <B1BUS> and <B3BUS>.

3.1.3 System Signals Output Function

TMP95C063F has the function to output system clock for CPU core and internal I/O (SCOUT) and bus start signal which indicates start of bus cycle (BS) for synchronizing to external circuit. SCOUT is the divided clock at the falling edge of external input clock X1. Figure 3.1 shows the timing of BS, SCOUT. See Figure 3.6 (1), (2), (3) about the timing inserting the wait.

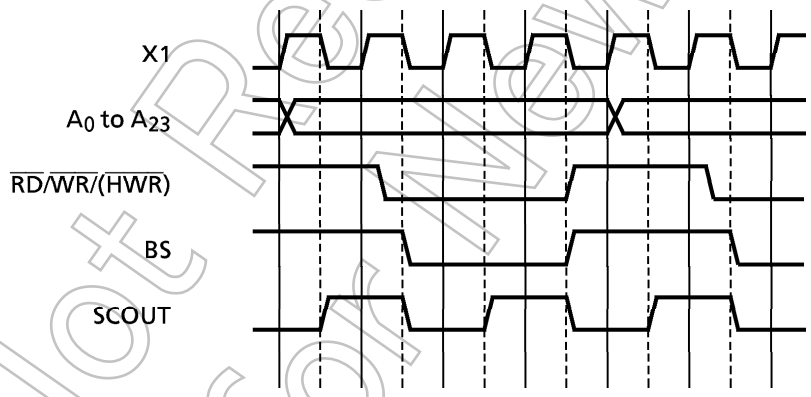
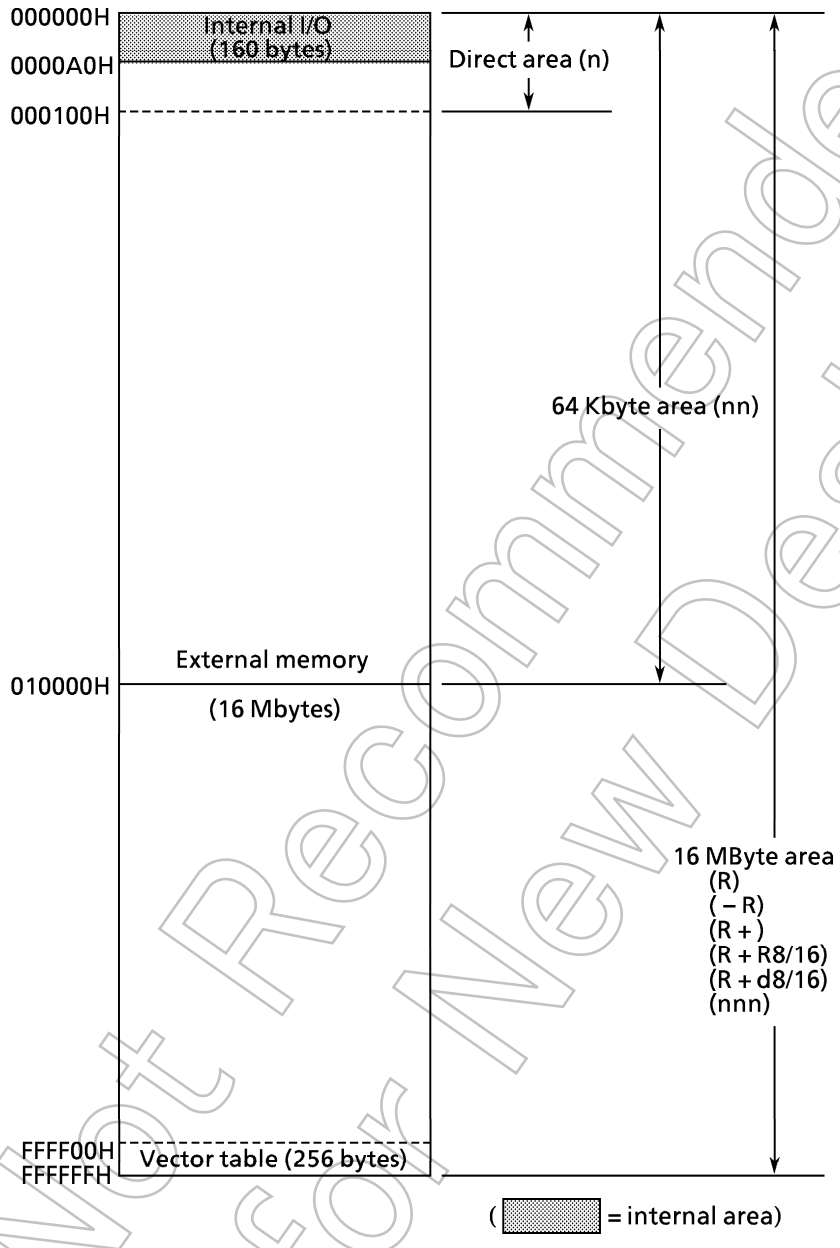


Figure 3.1 Timing Chart of BS, SCOUT (0 WAIT)

3.2 Memory Map

Figure 3.2 shows the TMP95C063 memory map.



Note : After reset, the stack pointer (XSP) is set to 100H.

Figure 3.2 Memory Map

3.2.1 Operation at internal I/O area access

TMP95C063 uses 160 bytes of address space (0H to 9FH) as an internal I/O area. Internal I/O registers are mapped on this area.

Operation of the internal I/O area access is different from that of the other address area access about following two points.

(1) In the internal I/O area access, $\overline{RD}$ and $\overline{WR}$ ($\overline{HWR}$) strobe signals are nonactive and fixed to high level.

However, in PSRAM mode set by P5<RDE>register, $\overline{RD}$ strobe signal becomes active also in the internal I/O area access. (See 3.5.3 Port5 (P52 to P57).)

(2) In the internal I/O area access, the number of waits becomes zero or one depending on the internal state of the CPU. This wait can't be controlled by chip select / wait controller (see 3.6 Chip Select / Wait Controller). When the specified address area overlaps with the internal I/O area, the operation as the internal I/O area takes priority of the specified address area.

3.3 Interrupts

TLCS-900 interrupts are controlled by the CPU interrupt mask flip-flops <IFF2 to 0> and the internal interrupt controller. Interrupts can come from a total of 35 sources (one pin is used for an external interrupt and internal I/O interrupt.):

- Interrupts from CPU itself: two (Software interrupt and illegal instructions)
- Interrupts from external pins ($\overline{\text{NMI}}$, $\overline{\text{NMI2}}$, INT0 to INT8): 11
- Interrupts from internal I/O: 18
- Interrupts from micro DMA: four

Individual interrupt vector numbers (fixed) are allocated to each interrupt source. Six levels of priority (variable) can be allocated to maskable interrupts. The priority of non-maskable interrupts is fixed at "7" (the highest priority).

When an interrupt is generated, the interrupt controller sends the priority value of that interrupt to the CPU. If more than one interrupt is generated simultaneously, the interrupt with the highest priority (7 non-maskable interrupts is the highest) is sent to the CPU.

The CPU compares the priority value with the value of the CPU interrupt mask register <IFF2 to 0>, and accepts the interrupt if the priority is higher or equal to the value in the CPU interrupt mask register. However, software interrupts and illegal instruction interrupts generated by the CPU are processed without comparison with the IFF <2:0> value.

The value of the interrupt mask register <IFF2 to 0> can be modified using the EI instruction (EI num sets IFF <2:0> to num). For example, executing "EI 3" enables acceptance of non-maskable interrupts and maskable interrupts with a priority of 3 or higher set in the interrupt controller.

The DI instruction (sets IFF <2:0> to "7") is operationally the same as specifying "EI 7". As maskable interrupts have priorities in the range of 0 to 6, the DI instruction disables acceptance of maskable interrupts. The EI instruction is valid immediately after its execution. With the TLCS-90, the EI instruction becomes valid only after the instruction following it is executed.

As well as the general-purpose interrupt processing mode described above, the TLCS-900 also supports micro DMA processing mode. In micro DMA mode, the CPU transfers data automatically, thus accelerating interrupt processing such as data transfer to, or from, internal I/Os.

In addition to using an interrupt to start a micro DMA request, TMP95C063 also supports the "software start function", which start micro DMA requests by software.

Figure 3.3 (1) is a flowchart of overall interrupt processing.

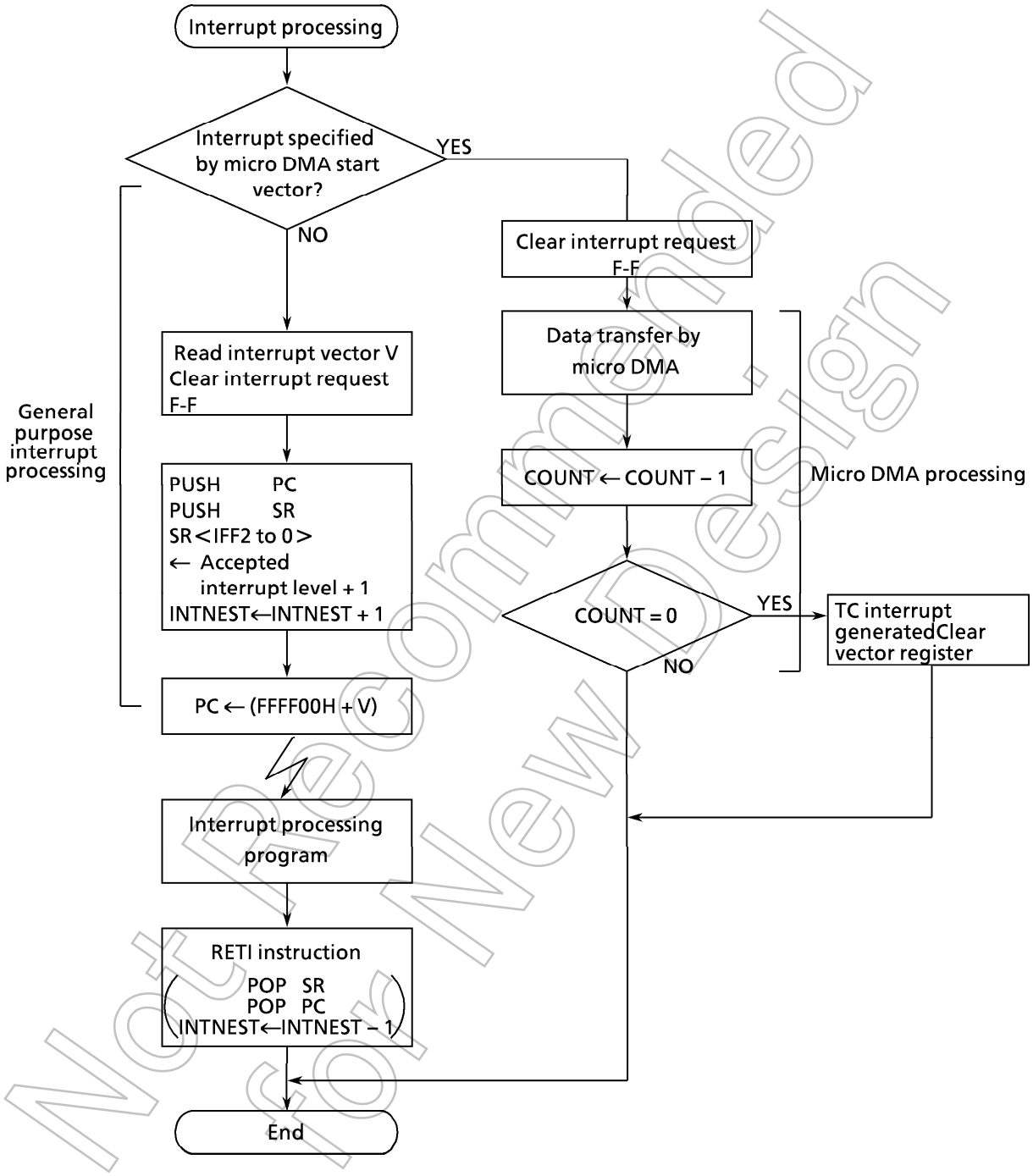


Figure 3.3 (1) Interrupt Processing Flowchart

3.3.1 General-Purpose Interrupt Processing

On receiving an interrupt, the CPU operates as follows

However, in the case of software interrupts and illegal instruction interrupts generated by the CPU, the CPU skips (1) and (3) and executes steps (2), (4) and (5).

(1) The CPU reads the interrupt vector from the interrupt controller. When more than one interrupt with the same level is generated at the same time, the interrupt controller generates an interrupt vector in accordance with the default priority (the smaller the vector value, the higher the priority (fixed)), and clears the interrupt request.

(2) The CPU pushes the program counter (PC) and status register (SR) onto the stack (the area pointed to by XSP).

(3) The CPU sets the interrupt mask register <IFF2 to 0> value to the level of the received interrupt incremented by 1. If the received interrupt is a level 7 interrupt, the CPU does not increment the interrupt mask register but sets it to "7".

(4) The CPU increments interrupt nesting counter INTNEST by 1.

(5) The CPU jumps to the address indicated by the data at address (FFFF00H + interrupt vector) and begins the interrupt processing routine.

The following table shows the times required by this processing.

Stack Area Bus Width	Interrupt Vector Area Bus Width	Number of Interrupt Processing Execution States
8	8	28
	16	24
16	8	22
	16	18

When interrupt processing is complete, the RETI instruction is executed to return processing to the main routine. Executing the RETI instruction restores the program counter (PC) and status register (SR) from the stack, and decrements interrupt nesting counter INTNEST by 1.

Non-maskable interrupts cannot be disabled by program. However, the program can enable or disable maskable interrupts, and can set priorities individually for each maskable interrupt source. The CPU accepts interrupt requests with a higher or equal priority than the value of the CPU interrupt mask register <IFF2 to 0>. On accepting an interrupt, the CPU sets the <IFF2 to 0> register to the received interrupt level incremented by 1. This means that if an interrupt is generated with a higher priority than the interrupt currently being processed, the CPU accepts the interrupt request for the higher priority interrupt and nests processing.

If a new interrupt request is generated while the CPU is accepting an interrupt and performing steps (1) to (5) described above, the CPU does not sample the new interrupt until after execution of the first instruction of the interrupt processing routine. Therefore, setting DI as the first instruction disables maskable interrupt nesting. (Note: The 900 and 900/L series sample the interrupt before executing the first instruction.)Resetting initializes the CPU mask register <IFF2 to 0> to "7". This disables maskable interrupts.

The area between addresses FFFF00H and FFFFFF (256 bytes) in TMP95C063 is assigned as the interrupt vector area.

The interrupt vector area varies with the product.

Table 3.3 (1) TMP95C063 Interrupt Table

Default Priority	Type	Interrupt Request Source	Vector "V"	Vector Reference Address	Micro DMA Start Vector
1		Reset, or SWI 0 instruction	0000H	FFFF00H	–
2		SWI 1 instruction	0004H	FFFF04H	–
3		INTUNDEF: Illegal instruction or SWI 2	0008H	FFFF08H	–
4	Non-	SWI 3 instruction	000CH	FFFF0CH	–
5	maskable	SWI 4 instruction	0010H	FFFF10H	–
6		SWI 5 instruction	0014H	FFFF14H	–
7		SWI 6 instruction	0018H	FFFF18H	–
8		SWI 7 instruction	001CH	FFFF1CH	–
9		NMI pin	0020H	FFFF20H	–
10		INTWD: Watchdog timer or NMI2 pin	0024H	FFFF24H	–
–		(HDMA)	–	–	–
11		INT0 pin	0028H	FFFF28H	0AH
12		INT1 pin	002CH	FFFF2CH	0BH
13		INT2 pin	0030H	FFFF30H	0CH
14		INT3 pin	0034H	FFFF34H	0DH
15		INT4 pin	0038H	FFFF38H	0EH
–		(reserved)	003CH	FFFF3CH	–
16		INT5 pin	0040H	FFFF40H	10H
17		INT6 pin	0044H	FFFF44H	11H
18		INT7 pin	0048H	FFFF48H	12H
19		INT8 pin	004CH	FFFF4CH	13H
20	Maskable	INTT0: 8-bit timer 0	0050H	FFFF50H	14H
21		INTT1: 8-bit timer 1	0054H	FFFF54H	15H
22		INTT2: 8-bit timer 2	0058H	FFFF58H	16H
23		INTT3: 8-bit timer 3	005CH	FFFF5CH	17H
24		INTT4: 8-bit timer 4	0060H	FFFF60H	18H
25		INTT5: 8-bit timer 5	0064H	FFFF64H	19H
26		INTT6: 8-bit timer 6	0068H	FFFF68H	1AH
27		INTT7: 8-bit timer 7	006CH	FFFF6CH	1BH
28		INTTR8: 16-bit timer 8 (TREG8)	0070H	FFFF70H	1CH
29		INTTR9: 16-bit timer 8 (TREG9)	0074H	FFFF74H	1DH
30		INTTRA: 16-bit timer 9 (TREGA)	0078H	FFFF78H	1EH
31		INTTRB: 16-bit timer 9 (TREGB)	007CH	FFFF7CH	1FH
32		INTRX0: Serial receive (channel.0)	0080H	FFFF80H	20H
33		INTTX0: Serial transmit (channel.0)	0084H	FFFF84H	21H
34		INTRX1: Serial receive (channel.1)	0088H	FFFF88H	22H
35		INTTX1: Serial transmit (channel.1)	008CH	FFFF8CH	23H
36		INTAD: A/D conversion complete	0090H	FFFF90H	24H
37		INTTC0: Micro DMA complete (channel.0)	0094H	FFFF94H	–
38		INTTC1: Micro DMA complete (channel.1)	0098H	FFFF98H	–
39		INTTC2: Micro DMA complete (channel.2)	009CH	FFFF9CH	–
40		INTTC3: Micro DMA complete (channel.3)	00A0H	FFFFA0H	–
–		Software Micro DMA	–	–	2FH

Setting reset or interrupt vector

① Reset vector

FFFF00H	PC (7:0)
FFFF01H	PC (15:8)
FFFF02H	PC (23:16)
FFFF03H	XX

② Interrupt vector (other than reset vector)

Vector reference address		
	+ 0	PC (7:0)
	+ 1	PC (15:8)
	+ 2	PC (23:16)
	+ 3	XX

XX : Don't care

(Setting example)

To define the reset vector as address 8100H, the $\overline{\text{NMI}}$ vector as address 9ABCH, and the INTAD vector as address 123456H:

```
ORG     8100H
LD      A, B
        :
ORG     9ABCH
LD      B, C
        :
ORG     123456H
LD      C, A
        :
ORG     0FFFF00H
DL      008100H         ; reset = 8100H

ORG     0FFFF20H
DL      009ABCH         ; NMI = 9ABCH

ORG     0FFFF90H
DL      123456H         ; INTAD = 123456H
```

Note:
ORG and DL are assembler directives.
ORG : For controlling location counter
DL : For defining long word data (32 bits)

3.3.2 Micro DMA

In addition to conventional interrupt processing, TMP95C063 supports the micro DMA function. For interrupt requests set for micro DMA, micro DMA processing is performed at the highest priority for maskable interrupts (level 6), regardless of the actual interrupt level set for the interrupt.

Because the function of micro DMA has been implemented with the cooperative operation of CPU, when CPU is a state of stand-by by HALT instruction, the requirement of micro DMA will be ignored (pending).

(1) Micro DMA Operation

When an interrupt request occurs for an interrupt specified by the micro DMA start vector register, micro DMA sends the micro DMA request to the CPU with the highest priority for maskable interrupts (level 6), regardless of the actual interrupt level set for the interrupt, and starts micro DMA. The micro DMA function has four channels. This allows micro DMA to be set for up to four interrupts at the same time.

When micro DMA is accepted, the interrupt request F-F for the micro DMA channel is cleared, data are automatically transferred from the transfer source address to the transfer destination address (the addresses are set in the control register), and the transfer count is decremented. If the decremented result is other than zero, micro DMA processing terminates. If the decremented result is zero, the CPU sends a micro DMA transfer end interrupt (INTTCn) to the interrupt controller, clears the micro DMA start vector register to 0, disables the next micro DMA startup, and terminates micro DMA processing.

If an interrupt request for the interrupt source used is received between the time that the micro DMA start vector is cleared and the time that it is reset, the CPU performs general-purpose processing at the specified interrupt level. Therefore, if the interrupt source is only being used for starting micro DMA (not used as an interrupt), set the interrupt level to zero.

When simultaneously using the same interrupt resource for both the micro DMA and general-purpose interrupts as described above, set the level of the interrupt source used to start micro DMA lower than the levels of all other interrupt sources. In this case, the cause of general interrupt is limited to the edge interrupt.

Example : When using timers 0 to 3 for running micro DMA 0 to 3
Set the interrupt level of timers 0 to 3 to 1
Set other interrupt levels to 2 to 6

Like other maskable interrupts, the priority of the micro DMA transfer end interrupt is determined by the interrupt level and default priority.
If multiple-channel micro DMA requests occur at the same time, the priority is determined by the channel numbers, not the interrupt levels. The lower the channel number, the higher the priority. (CH0 (high) → CH3 (low))

The transfer source and transfer destination addresses are set in 32-bit control registers. However, as only 24-bit addresses are output, the address space available to micro DMA is 16M bytes.

Three transfer modes are supported: 1-byte transfer, 1-word transfer (= two bytes), and 4-byte transfer. For each transfer mode, it is possible to specify whether to increment, decrement, or fix source and destination addresses after transfer. These modes facilitate data transfer from I/O to memory, from memory to I/O, and from I/O to I/O. For transfer mode details, see "Transfer Mode Register Details" later in this manual.

As a 16-bit transfer counter is used, micro DMA can perform a maximum of 65536 transfers (initializing the counter to 0000H specifies the maximum number of transfers).

The 26 interrupt sources with micro DMA start vectors (as listed in Table 3.3 (1)) can be used to start micro DMA processing. Together with the soft start function, this gives a total of 27 different micro DMA triggers.

Figure 3.3.2 (1) shows the micro DMA cycle for 1-word transfer in transfer destination address INC mode (the same apart from counter mode). (The conditions for this cycle are based on a 16-bit bus, 0 waits, and transfer source / transfer destination addresses both even-numbered values.).

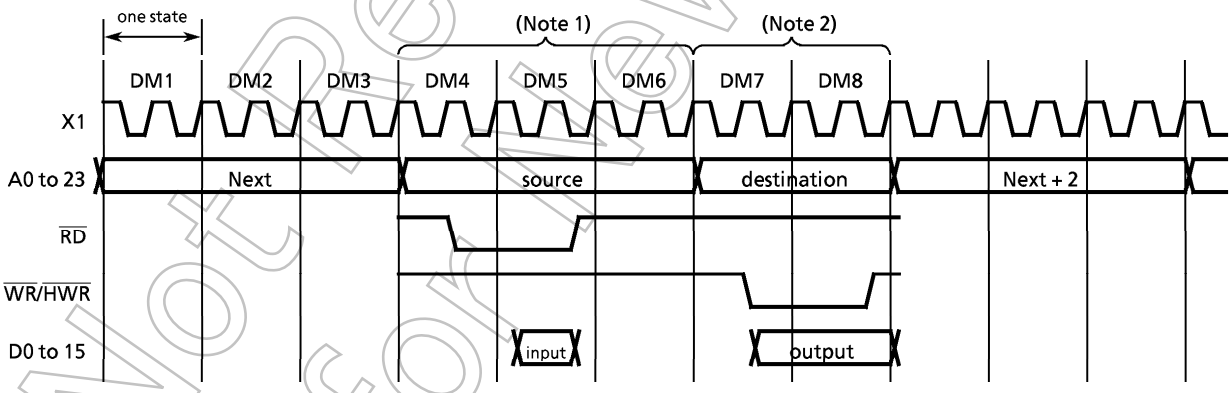


Figure 3.3.2 (1) Micro DMA Cycle Diagram

States 1-3 : Instruction fetch cycle (prefetches the next instruction code)
If the instruction cue buffer has three or more bytes of instruction code, the cycles are dummy cycles.

States 4-5 : Micro DMA read cycles

State 6 : Dummy cycle (address bus remains the same as in state 5)

States 6-8 : Micro DMA write cycle

Note 1 : If the source address area uses an 8-bit bus, two states are added.
If also the source address area uses a 16-bit bus and the source address is an odd-numbered address, two states are added.

Note 2 : If the destination address area uses an 8-bit bus, two states are added.
If also the destination address area uses a 16-bit bus and the destination address is an odd-numbered address, two states are added.

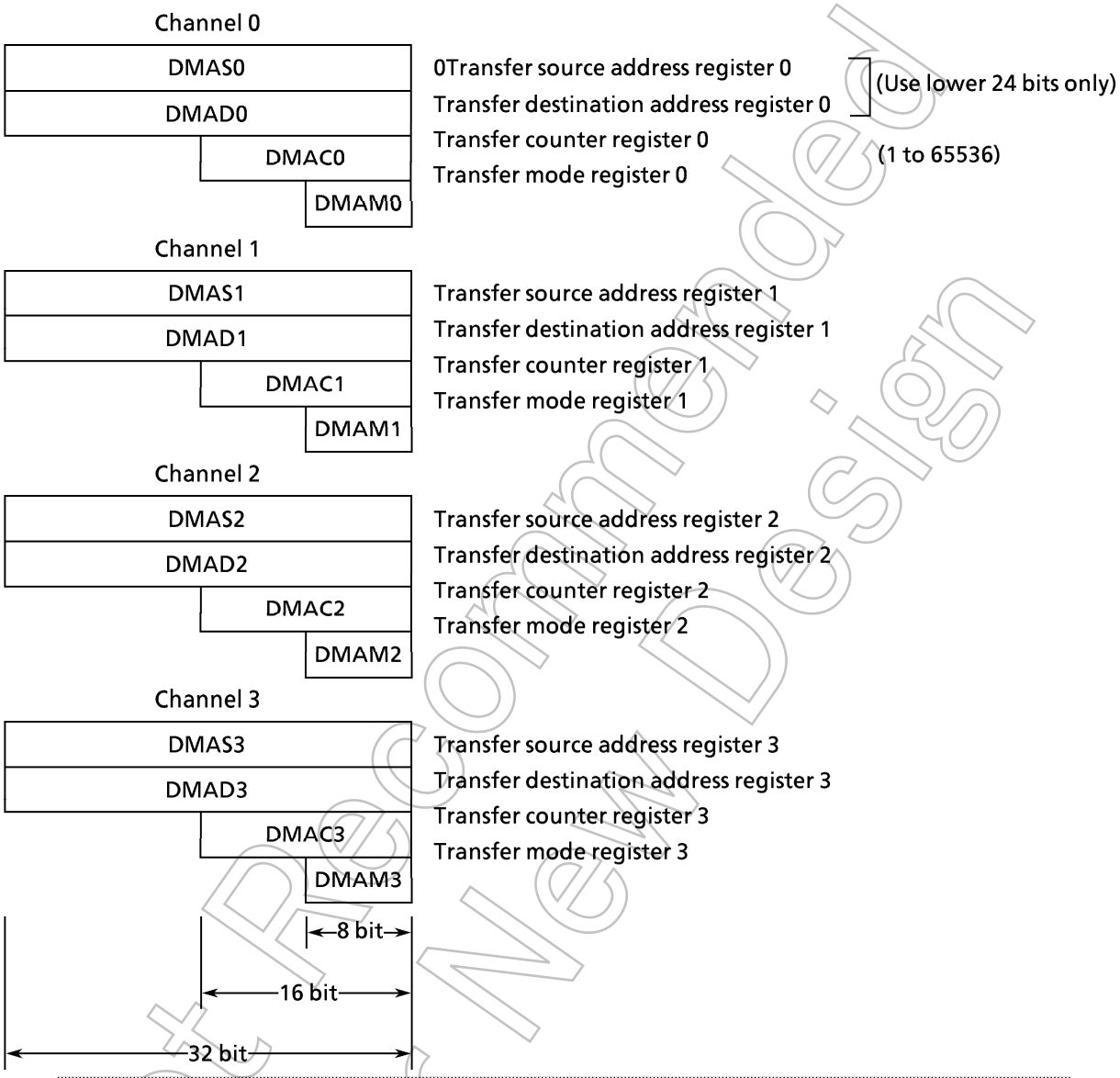
(2) Software Start Function

In addition to starting the micro DMA function by conventional interrupts, TMP95C063 includes a micro DMA software start function that starts micro DMA on the generation of the write cycle to the software DMA control register.

To trigger a software start, write the software micro DMA start vector "2FH" to the micro DMA start vector register (DMAxV). Next, writing data to the software DMA control register (SDMACRx) (regardless of the write value) causes micro DMA for the corresponding channel to run once. Writing again to the software DMA control register triggers another software start, provided the micro DMA transfer counter is set to other than "0". (It is not necessary to set the software micro DMA start vector again.)

Note that software start requests are one-shot requests and are not held over. If write cycle for the software DMA control register is generated when the software micro DMA start vector is not set, setting the software micro DMA start vector at a later time does not generate a software start.

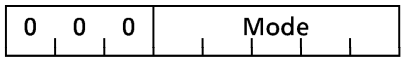
(3) Register Configuration (CPU Control Registers)



Data can be set in these control registers only with the "LDC cr, r" instruction.

(4) Transfer Mode Register Details

(DMAM0 to 3)



Note : When setting values in this register, set the upper three bits to 0.

Execution time : Minimum @ 25 MHz

ZZ : 0 = byte transfer, 1 = word transfer, 2 = 4-byte transfer, 3 = reserved

0 0 0 Z Z	Transfer destination address INC mode... For I/O to memory (DMADn +) ← (DMASn) DMACn←DMACn – 1 if DMACn = 0, then INTTC generated	8 states (640 ns) @ byte / word transfer 12 states (960 ns) @ 4-byte transfer
0 0 1 Z Z	Transfer destination address DEC mode... For I/O to memory (DMADn –) ← (DMASn) DMACn←DMACn – 1 if DMACn = 0, then INTTC generated	- 8 states (640 ns) @ byte / word transfer 12 states (960 ns) @ 4-byte transfer
0 1 0 Z Z	Transfer source address INC mode... For memory to I/O (DMADn) ← (DMASn +) DMACn←DMACn – 1 if DMACn = 0, then INTTC generated	8 states (640 ns) @ byte / word transfer 12 states (960 ns) @ 4-byte transfer
0 1 1 Z Z	Transfer source address DEC mode... For memory to I/O (DMADn) ← (DMASn –) DMACn←DMACn – 1 if DMACn = 0, then INTTC generated	8 states (640 ns) @ byte / word transfer 12 states (960 ns) @ 4-byte transfer
1 0 0 Z Z	Address fixed mode... For I/O to I/O (DMADn) ← (DMASn) DMACn←DMACn – 1 if DMACn = 0, then INTTC generated	8 states (640 ns) @ byte / word transfer 12 states (960 ns) @ 4-byte transfer
1 0 1 0 0	Counter mode... For interrupt count DMASn←DMASn + 1 DMACn←DMACn – 1 if DMACn = 0, then INTTC generated	5 states (400 ns)

(1 state = 80 ns @ 25 MHz)

Notes : n: Corresponding micro DMA channels 0-3
DMADn +/ DMASn+: Post-increment (increment the register value after transfer)
DMADn -/ DMASn-: Post-decrement (decrement the register value after transfer)
In the above table, "I/O" refers to fixed addresses and "memory" refers to incremented or decremented addresses.

Do not use undefined codes for transfer modes.

3.3.3 Interrupt Controller Operation

Figure 3.3.3 (1) is a block diagram of the interrupt circuit. The left-hand side of the diagram shows the interrupt controller circuit. The right-hand side shows the CPU interrupt request signal circuit and the halt release circuit.

For each interrupt channel (24 channels in total), the interrupt controller has an interrupt request flip-flop, an interrupt priority setting register, and a micro DMA start vector register. The interrupt request flip-flop latches interrupt requests from peripherals. The flip-flop is cleared to zero in the following cases: when reset occurs, when the CPU reads the channel vector of an interrupt it has received, when the CPU receives a micro DMA request (when micro DMA is set), and when an instruction that clears the interrupt for that channel is executed (by writing "0" to the clear bit in the interrupt priority setting register).

For example, to clear an INT0 interrupt request, set the register as shown below after executing the DI instruction.

Clears F-F to zero.

Reading the clear bit detects the state of the interrupt request flip-flop indicating the interrupt request state for the interrupt channel.

The interrupt priority can be set independently for each interrupt source by writing the priority to the interrupt priority setting register (eg, INTE0AD, INTE12). Six interrupt priorities from 1 to 6 are provided. Setting “0” (or “7”) disables the interrupt request. The priority of non-maskable interrupts ('NMI' pin, watchdog timer) is fixed at 7. If interrupt requests with the same level are generated at the same time, the default priority (the interrupt with the lowest priority or, in other words, the interrupt with the lowest vector value) is used to determine which interrupt request to accept first.

The interrupt controller sends the request with the highest priority and its vector address to the CPU. The CPU compares the value of the interrupt mask register in status register <IFF2 to 0> with the priority of the request signal and accepts the interrupt if the level of the request signal is higher. The CPU sets the received priority value incremented by 1 in status register <IFF2 to 0>. The CPU accepts only interrupt requests with a priority equal to or higher than this value during processing of the interrupt. On completion of interrupt processing (execution of the RETI instruction), the CPU restores the interrupt mask register value saved on the stack (the value before the interrupt) to CPU status register <IFF2 to 0>.

The interrupt controller also contains registers used to store the micro DMA start vectors (4 channels). These registers are I/O registers. Writing the start vectors (see Table 3.3 (1)) of the interrupt sources used to start micro DMA processing to the four register channels sets the corresponding interrupt requests as micro DMA requests. Before micro DMA processing, set values in the micro DMA parameter registers (eg, DMAS, DMAD).

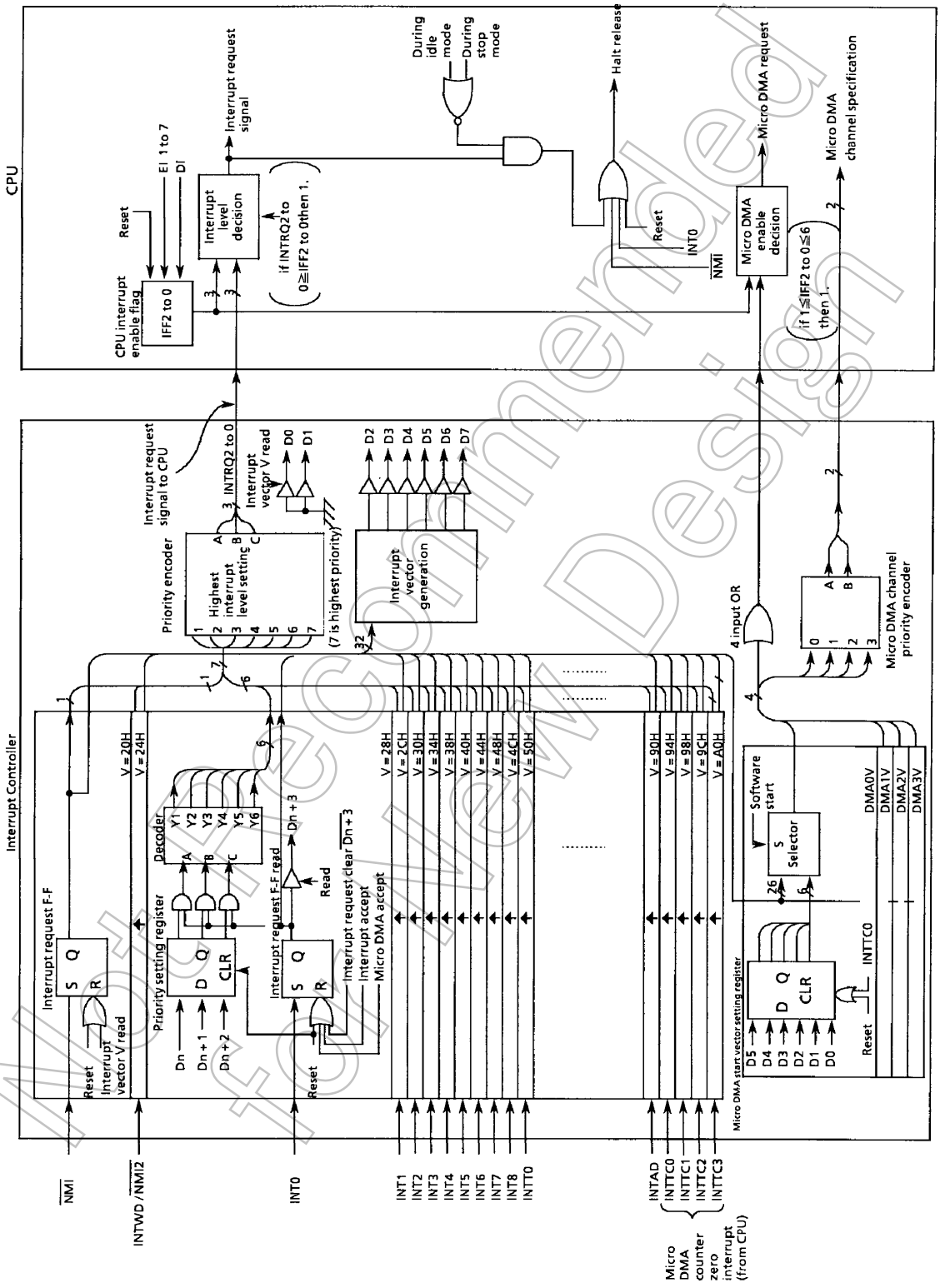


Figure 3.3.3 (1) Interrupt Controller Block Diagram

(1) Interrupt priority setting register

(Read-modify-write is inhibited.)

Symbol	Address	7	6	5	4	3	2	1	0
INTE0AD	70H	INTAD				INT0			
		IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTE12	71H	INT2				INT1			
		I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTE34	72H	INT4				INT3			
		I4C	I4M2	I4M1	I4M0	I3C	I3M2	I3M1	I3M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTE56	73H	INT6				INT5			
		I6C	I6M2	I6M1	I6M0	I5C	I5M2	I5M1	I5M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTE78	74H	INT8				INT7			
		I8C	I8M2	I8M1	I8M0	I7C	I7M2	I7M1	I7M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET01	75H	INTT1 (Timer 1)				INTT0 (Timer 0)			
		IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET23	76H	INTT3 (Timer 3)				INTT2 (Timer 2)			
		IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET45	77H	INTT5 (Timer 5)				INTT4 (Timer 4)			
		IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET67	78H	INTT7 (Timer 7)				INTT6 (Timer 6)			
		IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTET89	79H	INTTR9 (TREG9)				INTTR8 (TREG8)			
		IT9C	IT9M2	IT9M1	IT9M0	IT8C	IT8M2	IT8M1	IT8M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTETAB	7AH	INTTRB (TREGB)				INTTRA (TREGA)			
		ITBC	ITBM2	ITBM1	ITBM0	ITAC	ITAM2	ITAM1	ITAM0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0

←Interrupt source
←bit Symbol
←Read / Write
←Value after reset

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request.
0	0	1	Sets interrupt request level to 1.
0	1	0	Sets interrupt request level to 2.
0	1	1	Sets interrupt request level to 3.
1	0	0	Sets interrupt request level to 4.
1	0	1	Sets interrupt request level to 5.
1	1	0	Sets interrupt request level to 6.
1	1	1	Disables interrupt request.

IxxC	Function (Read)	Function (Write)
0	No interrupt request	Clears interrupt request flag.
1	Interrupt request	----- Don't care -----

Symbol	Address	7	6	5	4	3	2	1	0
INTES0	7BH	INTTX0				INTRX0			
		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTES1	7CH	INTTX1				INTRX1			
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTETC01	7DH	INTTC1				INTTC0			
		ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0
INTETC23	7EH	INTTC3				INTTC2			
		ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
		R/W	W			R/W	W		
		0	0	0	0	0	0	0	0

(2) External interrupt control

Interrupt input mode control register

IIMC (007FH)

	7	6	5	4	3	2	1	0
bit Symbol				NMI2E	IWDTS	I0IE	I0LE	NMIREE
Read/Write				W	W	W	W	W
Value after reset				0	0	0	0	0
Function				1: $\overline{\text{NMI2}}$ Input enable	0: WDT 1: $\overline{\text{NMI2}}$	1: INT0 Input enable	0: INT0 edge mode 1: INT0 level mode	1: $\overline{\text{NMI}}$ also functions at rising edge

Read-modify-write is inhibited.

Note : The INT0 pin can also be used for standby release (described below). When not using this pin for standby release, set this register to "0" to maintain port functions during standby.

$\overline{\text{NMI}}$ rising edge enable

0	Interrupt request generated on falling edge
1	Interrupt request generated on rising or falling edge

INT0 level enable

0	Rising edge detection interrupt
1	High level interrupt

INT0 input enable (note)

0	INT0 disable (P84 function only)
1	INT0 input enable

Watchdog timer $\overline{\text{NMI2}}$ select

0	Watchdog timer
1	$\overline{\text{NMI2}}$

$\overline{\text{NMI2}}$ Input enable

0	$\overline{\text{NMI2}}$ disable (P83 function only)
1	$\overline{\text{NMI2}}$ input enable

Setting of External Interrupt Pin Functions

Interrupt	Pin Name	Mode	Setting Method
$\overline{NMI}$	—	↓	IIMC<NMIREE> = 0
		↓↑ both falling and rising edge	IIMC<NMIREE> = 1
$\overline{NMI2}$	P83	↓ falling edge	IIMC<IWDTS> = 1, <NMI2E> = 1
INT0	P84	↑ rising edge	IIMC<I0LE> = 0, <I0IE> = 1
		level	IIMC<I0LE> = 1, <I0IE> = 1
INT1	P85	↑ rising edge	———
INT2	P86	↑ rising edge	———
INT3	P87	↑ rising edge	———
INT4	PB0	↑ rising edge	T8MOD<CAP12M1, 0> = 0, 0 or 0, 1 or 1, 1
		↓ falling edge	T8MOD<CAP12M1, 0> = 1, 0
INT5	PB1	↑ rising edge	———
INT6	PB4	↑ rising edge	T9MOD<CAP34M1, 0> = 0, 0 or 0, 1 or 1, 1
		↓ falling edge	T9MOD<CAP34M1, 0> = 1, 0
INT7	PB5	↑ rising edge	———
INT8	PD0	↑ rising edge	———

(3) Micro DMA start vector

This register assigns micro DMA processing to an interrupt source. The interrupt source with a micro DMA start vector that matches the vector set in this register is assigned as the micro DMA start source.

When the micro DMA transfer counter value reaches zero, the micro DMA transfer end interrupt corresponding to the channel is sent to the interrupt controller, the micro DMA start vector register is cleared, and the micro DMA start source of the channel is cleared. Therefore, to continue micro DMA processing, set the micro DMA start vector register again during the processing of the micro DMA transfer end interrupt.

If the same vector is set in the micro DMA start vector registers of more than one channel, the channel with the lowest number has a higher priority.

Accordingly, if the same vector is set in the micro DMA start vector registers of two channels, the interrupt generated in the channel with the lower number is executed until the micro DMA transfer is complete. If the micro DMA start vector of this channel is not set again, the next micro DMA is started for the channel with the higher number. (micro DMA chaining)

Micro DMA0 (Read-modify-write is inhibited.)

DMA0V (005AH)

	7	6	5	4	3	2	1	0
bit Symbol			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
Read/Write			W					
Value after reset			0	0	0	0	0	0
Function	Selects interrupt source allocated to Micro DMA channel 0.							

Micro DMA1 (Read-modify-write is inhibited.)

DMA1V (005BH)

	7	6	5	4	3	2	1	0
bit Symbol			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
Read/Write			W					
Value after reset			0	0	0	0	0	0
Function	Selects interrupt source allocated to Micro DMA channel 1.							

Micro DMA2 (Read-modify-write is inhibited.)

DMA2V (005CH)

	7	6	5	4	3	2	1	0
bit Symbol			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
Read/Write			W					
Value after reset			0	0	0	0	0	0
Function	Selects interrupt source allocated to Micro DMA channel 2.							

Micro DMA3 (Read-modify-write is inhibited.)

DMA3V (005DH)

	7	6	5	4	3	2	1	0
bit Symbol			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
Read/Write			W					
Value after reset			0	0	0	0	0	0
Function	Selects interrupt source allocated to Micro DMA channel 3.							

(4) Notes

The instruction execution unit and the bus interface unit of this CPU operate independently. Therefore, immediately before an interrupt is generated, if the CPU fetches an instruction that clears the corresponding interrupt request flag, the CPU may execute the instruction that clears the interrupt request flag between accepting and reading the interrupt vector.

To avoid the above problem, place instructions that clear interrupt request flags after a DI instruction. In the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing instruction and following more than one instruction are executed. When EI instruction is placed immediately after clearing instruction, an interrupt becomes enable before interrupt request flags are cleared.

In the case of changing the value of the interrupt mask register<IFF2 to 0> by execution of POP SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.

In addition, take care as the following three circuits are exceptional and demand special attention.

INT0 level mode	INT0 in level mode is not an edge-detect interrupt, so the interrupt request flip-flop function is canceled. The peripheral interrupt request bypasses the S input of the flip-flop, and acts as the Q output. Changing modes from edge to level automatically clears the interrupt request flag.
	If the CPU enters the interrupt response sequence as a result of setting INT0 from 0 to 1, INT0 must be held at 1 until the interrupt response sequence is completed. If the INT0 level mode is used to release a halt, INT0 must be held at 1 from the time INT0 changes from 0 to 1, to the time when the halt is released. (Ensure that INT0 does not go back 0 due to noise before the halt is released.)
	When switching modes from level to edge, any interrupt request flag set in level mode is not cleared. Accordingly, clear the interrupt request flag using the following sequence.
	DI
	LD (IIMC), 00H ; Switches from level to edge.
	LD (INTE0AD), 00H ; Clears interrupt request flag.
	EI

INTAD	The interrupt request flip-flop can only be cleared by reset or by reading the A/D conversion result register, not by an instruction.
INTRX	The interrupt request flip-flop can only be cleared by reset or by reading the serial channel receive buffer, not by an instruction.

Note : The following instructions or pin changes are equivalent to instructions that clear the interrupt request flag.

INT0 : Instructions that switch to level mode after an interrupt request is generated in edge mode.
The pin input changes from high to low after an interrupt request is generated in level mode. ("H" → "L")

INTAD : Instructions that read the A/D conversion result register.

INTRX : Instructions that read the receive buffer.

3.4 Standby Function

Executing the HALT instruction sets either RUN, IDLE, or STOP mode depending on the content of WDMOD<HALTM1:0>.

(1) RUN : Halts the CPU only. Power dissipation remains almost unchanged.

(2) IDLE : Operates only the internal oscillator, while halts all other circuits.

(3) STOP : Halts all internal circuits, including the internal oscillator.

These halt states are released depending on the mode. For details, see Table 3.4 (2). (Note: Halt cannot be released by triggering micro DMA except for INT0.)

Example of releasing halt.

On execution of the HALT instruction, the device enters standby state in RUN mode. Release halt using INT0.

```
Address
8203H    LD    (IIMC), 04H          ; Enables INT0 input.
8206H    LD    (INTE0AD), 06H       ; Sets INT0 to level 6
8209H    EI    5                    ; Sets CPU interrupt accept level to 5.
820BH    LD    (WDMOD), 00H         ; Sets RUN mode.
820EH    HALT                       ; Halts CPU.
```


```
820FH    LD    XX, XX               RETI
```

(1) RUN mode

Figure 3.4 (1) is the timing chart for releasing a halt in RUN mode using an interrupt.

In RUN mode, the MCU internal system clock does not stop after the HALT instruction is executed. Only CPU instruction execution stops. Therefore, the CPU performs repeated dummy cycles until the halt state is released. In the halt state, interrupt requests are sampled on the falling edge of the CLK signal.

The halt state can only be released by external interrupts (INT1 to 8, $\overline{\text{NMI2}}$) in RUN mode.

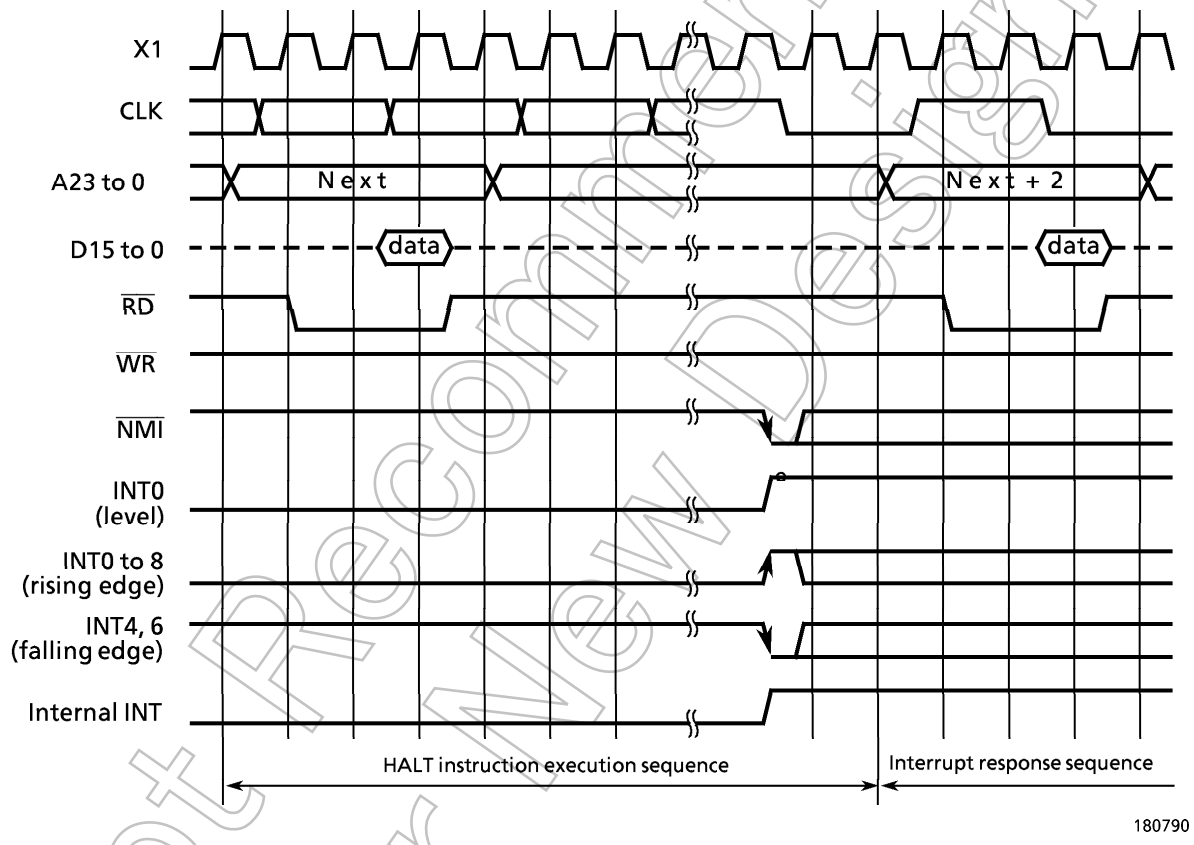


Figure 3.4 (1) Timing Chart for Releasing Halt in RUN Mode Using Interrupt

(2) IDLE mode

Figure 3.4 (2) is the timing chart for releasing a halt in IDLE mode using an interrupt.

In IDLE mode, the MCU internal system clock stops. Only the internal oscillator functions. The CLK pin is fixed at "1".

In the halt state, interrupt requests are sampled asynchronously to the system clock. The release from the halt state (operation restart), however, is synchronized with the clock.

In IDLE mode, interrupt requests other than external interrupts ($\overline{\text{NMI}}$, INT0) are disabled. (Note: The halt state in IDLE mode cannot be released by $\overline{\text{NMI2}}$.)

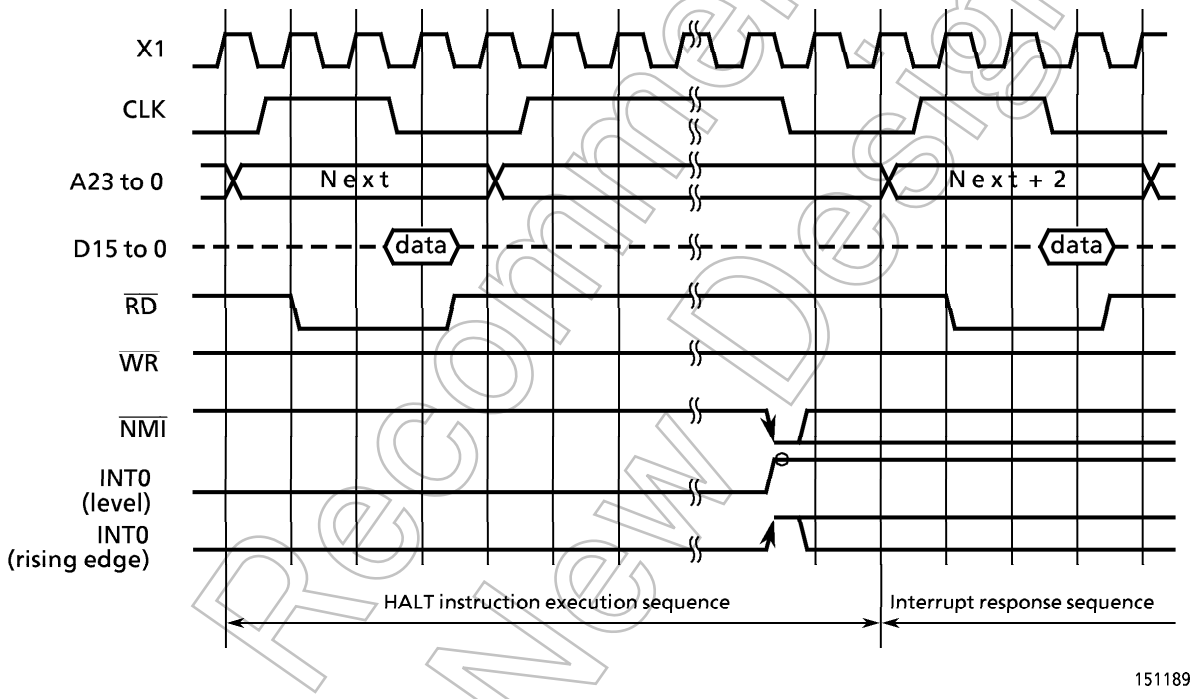


Figure 3.4 (2) Timing Chart for Releasing Halt in IDLE Mode Using Interrupt

(3) STOP mode

Figure 3.4 (3) is the timing chart for releasing a halt in STOP mode using an interrupt.

In STOP mode, all internal circuits stop, including the internal oscillator. Also, in STOP mode, all pins, apart from a few exceptions, are set to high impedance and are disconnected from the internal circuit of the MCU.

However, setting WDMOD<DRVE> in the internal I/O register to "1" specifies that pins maintain the states prior to the halt. Reset clears the register to "0".

When the CPU receives an interrupt request, the internal oscillator restarts. Then, after the time set by the warm-up counter for the internal oscillation to stabilize, the system clock starts its output. The WDMOD<WARM> bit sets the warm-up time. Setting this bit to 0 specifies a warm-up time of 2^{14} clock cycles; setting the bit to 1 specifies a warm-up time of 2^{16} clock cycles. Reset clears WDMOD<WARM> to 0.

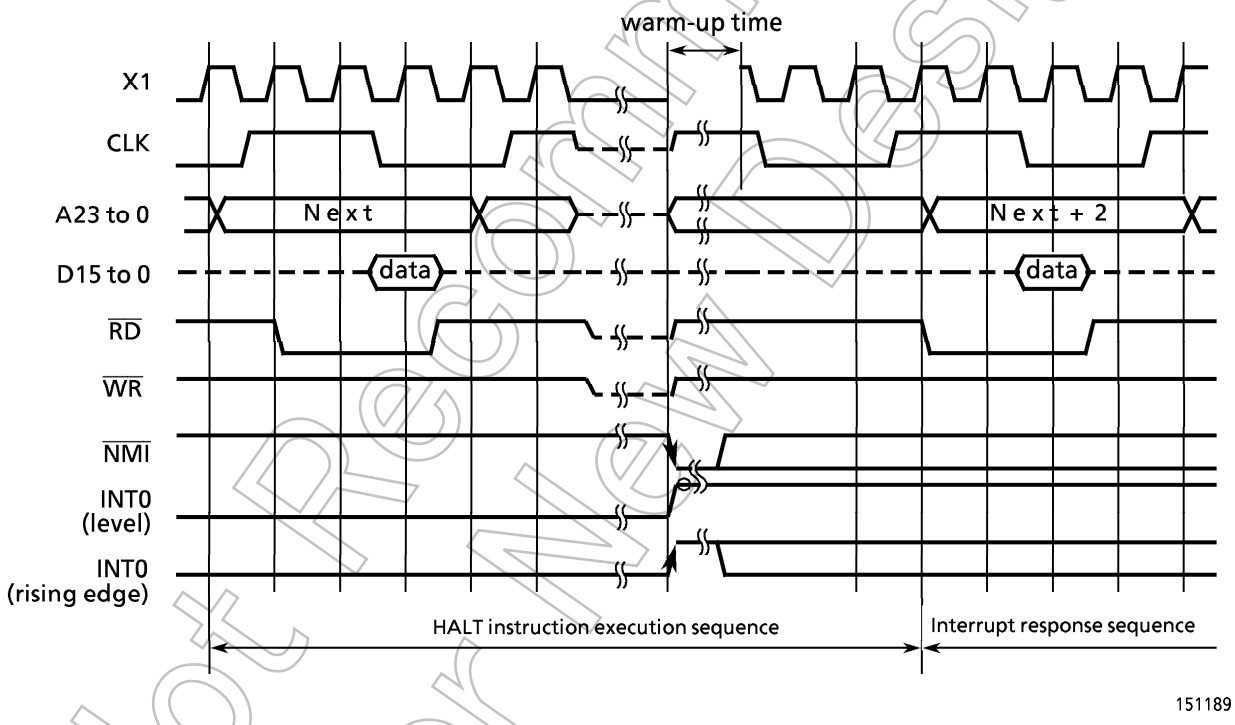


Figure 3.4 (3) Timing Chart for Releasing Halt in STOP Mode Using Interrupt

STOP mode can only be released by an NMI pin or INT0 pin interrupt, or by reset. When STOP mode is released by other than reset, the system clock starts its output after the time set by the warm-up counter for the internal oscillation to stabilize. When using reset to release stop mode, input reset signals long enough for stable oscillation.

In systems with an external oscillator, the warm-up counter also operates when STOP mode is released. Therefore, such systems also require a warm-up time between input of release signal and system clock output.

Note: Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, $\overline{\text{NMI2}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE or STOP mode. (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

Figure 3.4 (1) Pin states in STOP Mode

Pin Name	Input / Output	DRVE = 0	DRVE = 1
D0 to D7	I/O	HI-Z*	HI-Z*
P10 to P17 (D8 to D15)	Input mode (P10 to P17) Output mode (P10 to P17) I/O (D8 to D15)	HI-Z* HI-Z* HI-Z*	HI-Z* Output HI-Z*
P20 to P27 (A16 to A23)	Output	HI-Z	Output
A0 to A15	Output	HI-Z	Output
$\overline{RD}$, $\overline{WR}$	Output	HI-Z	"1"
P52 to P55 ($\overline{HWR}$, $\overline{BUSRQ}$, $\overline{BUSAK}$, R/$\overline{W}$)	Input mode Output mode	PU* PU*	PU△ Output
P56, P57 ($\overline{CS0}$, $\overline{CS2}$)	Output	HI-Z	Output
P60 to P67 (DRAM control signal)	Output	HI-Z	Output
P70 to P77 (PG00 to 03) (PG10 to 13)	Input mode Output mode	PU* PU*	PU△ Output
P80 to P82 (BS, SCOUT, $\overline{WAIT}$)	Input mode Output mode	PU* PU*	PU△ Output
P83 to P87 ($\overline{NMI2}$, INT0 to 3) PD0 (INT8)	Input mode Output mode	PU△ PU△	PU△ Output
P90 to P97 (TI0, 2, 4, 6, TO1, 3, 5, 7)	Input mode Output mode	PU* PU*	PU△ Output
PA0 to PA7 (TXD, RXD, $\overline{CTS}$, SCLK)	Input mode Output mode	PU* PU*	PU△ Output
PB0 to PB7 (TI8 to B, TO8 to B, INT4 to 7)	Input mode Output mode	PU* PU*	PU△ Output
PC0 to PC7 (AN0 to 7)	Input (PORT) Input (AN0 - AN7)	Invalid ◎	Invalid ◎
PD1 to PD4	Input mode Output mode	PU* PU*	PU△ Output
PE0 to PE7	Input mode Output mode	PU* PU*	PU△ Output
$\overline{NMI}$	Input	valid	valid
$\overline{WDTOUT}$	Output	Output	Output
CLK	Output	HI-Z	"1"
$\overline{RESET}$	Input	valid	valid
AM (8/$\overline{16}$)	Input	◎	◎
$\overline{EA}$	Input	◎	◎
X1	Input	Invalid	Invalid
X2	Output	"1"	"1"

Output : Maintains output states prior to a halt.
PU : Programmable pull-up pin. During a halt, pulled up by setting.
* : The input gate is disabled.
No through current, even at high impedance.
An instruction to access the port register (Ex. P8) should not be placed before the HALT instruction. There is possibility that the input gate is not disabled.
△ : When the pin is set to high impedance by disconnecting the pull-up resistor, the input gate continues to operate. Therefore, fix the pin state to prevent current flow.
◎ : Must be driven externally.
Valid : Input is valid.
Invalid : Input is invalid. As the input gate is disabled, no through current.

Table 3.4 (2) I/O Operation During Halt and Halt and Release

Halt Mode		RUN	IDLE	STOP
WDMOD 〈 HALTM1, 0 〉		00	10	01
Operation Block	CPU	Halt		
	I/O Port	Operation		See Table 3.4 (1)
	8-bit timer		Halts	
	8-bit PWM timer			
	16-bit timer			
	pattern generator			
	Serial interface			
	A/D converter			
	D/A converter			
	Watchdog timer			
	DRAM controller			
	Interrupt controller			

Interrupt mask and request level settings			Interrupt request level ≧ interrupt mask <IFF2-0>			Interrupt request level *2 < interrupt mask <IFF2-0>		
Halt Mode			RUN	IDLE	STOP	RUN	IDLE	STOP
Halt Release Source	Interrupt	NMI	◎	◎	◎*1	◎	◎	◎*1
		NMI2	◎	×	×	◎	×	×
		INTWD	◎	×	×	◎	×	×
		INT0	◎	◎	◎*1	○	○	○*1
		INT1-8	◎	×	×	×	×	×
		INTT0-7	◎	×	×	×	×	×
		INTTR8-B	◎	×	×	×	×	×
		INTRXD0, 1	◎	×	×	×	×	×
		INTTXD0, 1	◎	×	×	×	×	×
		INTAD	◎	×	×	×	×	×
	RESET		◎	◎	◎	◎	◎	◎

◎ : After a halt is released, interrupt processing begins. (Reset initializes the LSI.)
○ : After a halt is released, processing begins from the next address following the HALT instruction.
× : Cannot be used to release a halt.
*1 : Halt is released after the warm-up time has elapsed.
*2 : Same as a DI instruction.

3.5 Port Functions

TMP95C063 has a total of 91 I/O Port pins when the AM8/16 pin is "1", and 83 I/O Port pins when the AM8/$\overline{16}$ pin is "0".

In addition to Functioning as general-purpose I/O Ports, these pins are also used by internal CPU and I/O Functions. Table 3.5 (1) lists pin Functions. Table 3.5 (2) lists Port setting.

Table 3.5 (1) Port Functions (R: ↑ = programmable pull-up resistor connected, ↓ = programmable pull-down resistor connected)

Port Name	Pin Name	Number of Pins	I/O	R	I/O Setting Unit	Pin Name when used by internal Function
1	P10 to P17	8	I/O	–	bit	D8 to D15
Port 2	P20 to P27	8	Output	–	(fixed)	A16 to A23
Port 5	P52	1	I/O	↑	bit	$\overline{HWR}$
	P53	1	I/O	↑	bit	$\overline{BUSRQ}$
	P54	1	I/O	↑	bit	$\overline{BUSAK}$
	P55	1	I/O	↑	bit	R/$\overline{W}$
	P56	1	Output	–	(fixed)	$\overline{CS0}$
	P57	1	Output	–	(fixed)	$\overline{CS2}$
Port 6	P60	1	Output	–	(fixed)	$\overline{CS1}$ / $\overline{RAS1}$
	P61	1	Output	–	(fixed)	$\overline{CAS1}$ / $\overline{WE1}$
	P62	1	Output	–	(fixed)	$\overline{LCAS1}$ / $\overline{LW1}$ / $\overline{REFOUT1}$
	P63	1	Output	–	(fixed)	$\overline{UCAS1}$ / $\overline{UW1}$ / $\overline{WE1}$
	P64	1	Output	–	(fixed)	$\overline{CS3}$ / $\overline{RAS3}$
	P65	1	Output	–	(fixed)	$\overline{CAS3}$ / $\overline{WE3}$
	P66	1	Output	–	(fixed)	$\overline{LCAS3}$ / $\overline{LW3}$ / $\overline{REFOUT3}$
	P67	1	Output	–	(fixed)	$\overline{UCAS3}$ / $\overline{UW3}$ / $\overline{WE3}$
Port 7	P70 to P77	8	I/O	↑	bit	PG00 to 03, PG10 to 13
Port 8	P80	1	I/O	↑	bit	BS
	P81	1	I/O	↑	bit	Scout
	P82	1	I/O	↑	bit	$\overline{WAIT}$
	P83	1	I/O	↑	bit	$\overline{NMI2}$
	P84	1	I/O	↑	bit	INT0
	P85	1	I/O	↑	bit	INT1
	P86	1	I/O	↑	bit	INT2
	P87	1	I/O	↑	bit	INT3
Port 9	P90	1	I/O	↑	bit	TI0
	P91	1	I/O	↑	bit	TO1
	P92	1	I/O	↑	bit	TI2
	P93	1	I/O	↑	bit	TO3
	P94	1	I/O	↑	bit	TI4
	P95	1	I/O	↑	bit	TO5
	P96	1	I/O	↑	bit	TI6
	P97	1	I/O	↑	bit	TO7

Table 3.5 (1) Port Functions (R: ↑ = programmable pull-up resistor attached, ↓ = programmable pull-down resistor attached)

Port Name	Pin Name	Number of Pins	I/O	R	I/O Setting Unit	Pin Name when used by internal Function
Port A	PA0	1	I/O	↑	bit	TXD0
	PA1	1	I/O	↑	bit	RXD0
	PA2	1	I/O	↑	bit	$\overline{CTS0}$
	PA3	1	I/O	↑	bit	SCLK0
	PA4	1	I/O	↑	bit	TXD1
	PA5	1	I/O	↑	bit	RXD1
	PA6	1	I/O	↑	bit	$\overline{CTS1}$
	PA7	1	I/O	↑	bit	SCLK1
Port B	PB0	1	I/O	↑	bit	TI8/INT4
	PB1	1	I/O	↑	bit	TI9/INT5
	PB2	1	I/O	↑	bit	TO8
	PB3	1	I/O	↑	bit	TO9
	PB4	1	I/O	↑	bit	TIA/INT6
	PB5	1	I/O	↑	bit	TIB/INT7
	PB6	1	I/O	↑	bit	TOA
	PB7	1	I/O	↑	bit	TOB
Port C	PC0 to PC7	8	Input	–	(fixed)	AN0 to AN7
Port D	PD0	1	I/O	↑	bit	INT8
	PD1 to PD4	4	I/O	↑	bit	
Port E	PE0 to PE7	8	I/O	↑	bit	

Table 3.5 (2) I/O Port Setting

Port	Pin Name	Port (I/O) or Function	I/O Register		
			Pn	PnCR	PnFC
Port1	P1 (0 : 7) (Note 1)	Input Port	X	0	–
		Output Port	X	1	
		D (8 : 15)	X	X	
Port2	P2 (0 : 7)	Output Port	X	–	0
		A (16 : 23)	X		1
Port5	$\overline{RD}$	$\overline{RD}$ Output only for External Access	1	–	–
		Always $\overline{RD}$ Output	0		
	P5 (2 : 5)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	P52	$\overline{HER}$ Output	X	1	1
	P53	$\overline{BUSRQ}$ Input (no pull-up)	0	0	1
		$\overline{BUSRQ}$ Input (with pull-up)	1	0	1
	P54	$\overline{BUSAK}$ Output	X	1	1
	P55	R / $\overline{W}$ Output	X	1	1
	P5 (6 :7)	Output Port	X	–	0
	P56	$\overline{CS0}$ Output	X		1
	P57	$\overline{CS2}$ Output	X		1
Port6 (Note 2)	P6 (0 : 5)	Output Port	X	–	0
	P60	$\overline{CS1}$ / $\overline{RAS1}$ Output	X		1
	P61	$\overline{CAS1}$ / $\overline{WE1}$ Output	X		1
	P62	$\overline{LCAS1}$ / $\overline{LW1}$ / $\overline{REFOUT1}$ Output	X		1
	P63	$\overline{UCAS1}$ / $\overline{UW1}$ / $\overline{WE1}$ Output	X		1
	P64	$\overline{CS3}$ / $\overline{RAS3}$ Output	X		1
	P65	$\overline{CAS3}$ / $\overline{WE3}$ Output	X		1
	P66	$\overline{LCAS3}$ / $\overline{LW3}$ / $\overline{REFOUT3}$ Output	X		1
	P67	$\overline{UCAS3}$ / $\overline{UW3}$ / $\overline{WE3}$ Output	X		1
Port7	P7 (0 : 7)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
		PGn Output	X	1	1

Note 1: Function is fixed according to Input to AM8 / $\overline{16}$ pin.

Note 2: The Function of P60 or P64 (CS/RAS) is selected using B1CS or B3CS Register. For the DRAM control Functions, see Table 3.7 (1) DRAM control pins.

Table 3.5 (2) I/O Port Setting

Port	Pin Name	Port (I/O) or Function	I/O Register		
			Pn	PnCR	PnFC
Port8	P8 (0 : 7)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	P80	BS Output	X	1	1
	P81	SCOUT Output	X	1	1
	P82	$\overline{\text{WAIT}}$ Input (no pull-up)	0	0	–
		$\overline{\text{WAIT}}$ Input (with pull-up)	1	0	
	P83 (Note 3)	$\overline{\text{NMI2}}$ Input (no pull-up)	0	0	
		$\overline{\text{NMI2}}$ Input (with pull-up)	1	0	
	P84 (Note 3)	INT0 Input (no pull-up)	0	0	
		INT0 Input (with pull-up)	1	0	
	P85	INT1 Input (no pull-up)	0	0	
		INT1 Input (with pull-up)	0	0	
	P86	INT2 Input (no pull-up)	1	0	
		INT2 Input (with pull-up)	0	0	
	P87	INT3 Input (no pull-up)	1	0	
		INT3 Input (with pull-up)	1	0	
Port9	P9 (0 : 7)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	P90	TI0 Input (no pull-up)	0	0	–
		TI0 Input (with pull-up)	1	0	
	P92	TI2 Input (no pull-up)	0	0	
		TI2 Input (with pull-up)	1	0	
	P94	TI4 Input (no pull-up)	0	0	
		TI4 Input (with pull-up)	1	0	
	P96	TI6 Input (no pull-up)	0	0	
		TI6 Input (with pull-up)	0	0	
	P91	TO1 Output	X	0	1
	P92	TO3 Output	X	0	1
	P95	TO5 Output	X	0	1
	P97	TO7 Output	X	0	1

Note 3: When P83 / P84 pin is used as $\overline{\text{NMI2}}$ / INT0 pin, set IIMC <I0IE> / <NMI2E> to "1". (Input enable)

Port	Pin Name	Port (I/O) or Function	I/O Register		
			Pn	PnCR	PnFC
Port A	PA (0 : 7)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	PA0	TXD0 Output	X	1	1
	PA4	TXD1 Output	X	1	1
	PA1	RXD0 Input (no pull-up)	0	0	–
		RXD0 Input (with pull-up)	1	0	
	PA5	RXD1 Input (no pull-up)	0	0	
		RXD1 Input (with pull-up)	1	0	
	PA2	$\overline{CTS0}$ Input (no pull-up)	0	0	
		$\overline{CTS0}$ Input (with pull-up)	1	0	
	PA6	$\overline{CTS1}$ Input (no pull-up)	0	0	
		$\overline{CTS1}$ Input (with pull-up)	1	0	
	PA3	SCLK0 Output Port	X	1	1
		SCLK0 Input Port (no pull-up)	0	0	0
		SCLK0 Input Port (with pull-up)	1	0	0
	PA7	SCLK1Output Port	X	1	1
		SCLK1Input Port (no pull-up)	0	0	0
		SCLK1Input Port (with pull-up)	1	0	0
Port B	PB (0 : 7)	Input Port (no pull-up)	0	0	0
		Input Port (with pull-up)	1	0	0
		Output Port	X	1	0
	PB0	TI8 / INT4 Input (no pull-up)	0	0	–
		TI8 / INT4 Input (pull-up)	1	0	
	PB1	TI9 / INT5 Input (no pull-up)	0	0	
		TI9 / INT5 Input (pull-up)	1	0	
	PB4	TIA / INT6 Input (no pull-up)	0	0	
		TIA / INT6 Input (pull-up)	1	0	
	PB5	TIB / INT7 Input (no pull-up)	0	0	
		TIB / INT7 Input (pull-up)	1	0	
	PB2	TO8 Output	X	1	1
	PB3	TO9 Output	X	1	1
	PB6	TOA Output	X	1	1
	PB7	TOB Output	X	1	1
Port C	PC (0 : 7)	Input Port	X	–	
		AN (0 : 7) Input (Note 4)	X		
Port D	PD (0 : 4)	Input (no pull-up)	0	0	0
		Input (with pull-up)	1	0	0
		Output Port	X	1	0
Port E	PE (0 : 7)	Input (no pull-up)	0	0	0
		Input (with pull-up)	1	0	0
		Output Port	X	1	0

Note 4: Select the Input channels for the A/D converter in ADMOD2<ADCHn>.

3.5.1 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O Port. Bits can be individually set as either Inputs or Outputs by control Register P1CR. A reset clears all bits of the Output latches in the Port 1 Register (P1) and P1CR Registers to "0", setting Port 1 to an Input Port.

In addition to Functioning as a general-purpose I/O Port, Port 1 can also Function as data bus (D8 to 15).

TMP95C063 determines the Port Function and the data bus Function according to the Input state of AM8/$\overline{16}$ pin after reset. When AM8/$\overline{16}$ is set to low level, the data bus Functions. When AM8/$\overline{16}$ is set to high level, the Port Functions. When using as the data bus (AM8/$\overline{16}$ = "0"), the bit of P1CR Register should not be set to 1.

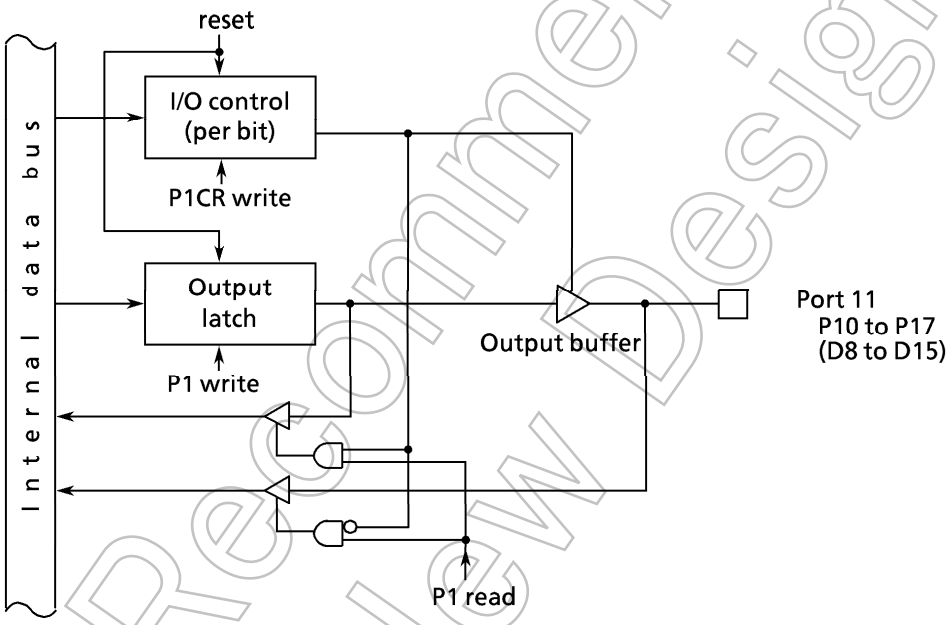


Figure 3.5 (1) Port 1

Port 1 Register

P1 (0001H)	7	6	5	4	3	2	1	0
bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R/W							
After reset	input mode (all bits of output latches are cleared to zero)							

Port 1 Control Register

P1CR (0004H)	7	6	5	4	3	2	1	0
bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN 1 : OUT							

Read-modify-write is prohibited for registers P1CR.

Port 1 function setting

P1CR <P1XC> \ AM8/16	0	1
0	data bus (D15 to 8)	input port
1	Don't set	output port

Note: <PIXC> is bit X of the P1CR register.

Figure 3.5 (2) Port 1 Registers

3.5.2 Port 2 (P20 to P27)

Port 2 is an 8-bit general-purpose output-only port. A reset sets all bits of the output latches in the port 2 register (P2) to "1" and all port pins output "1".

In addition to functioning as a general-purpose output port, port 2 can also function as address bus (A16 to 23). The port function is specified by function register P2FC. Port pins can be selected individually as either output ports or address bus pins.

In TMP95C063 with external ROM, a reset sets all bits of the function register to "1", and sets the pins as address bus pins (A16 to A23).

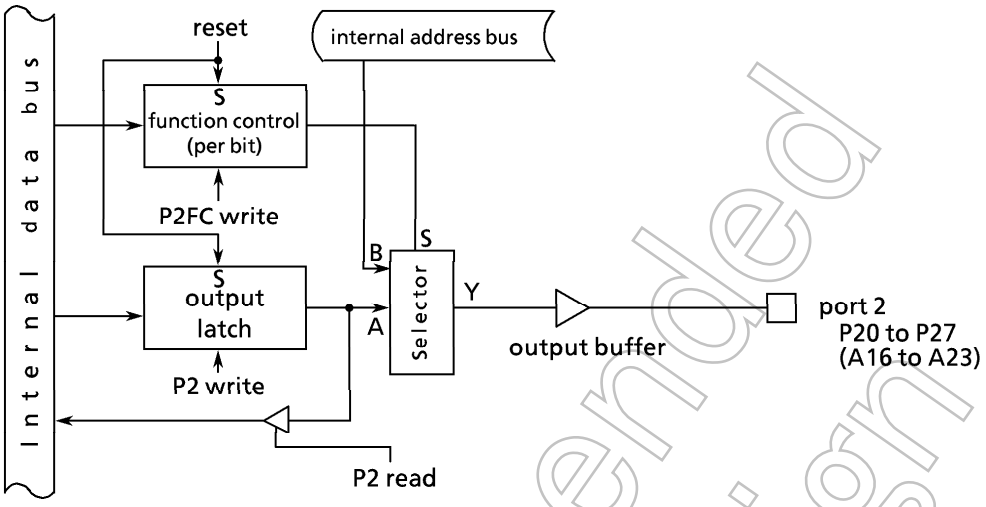


Figure 3.5 (3) Port 2

Port 2 Register

P2 (0006H)

	7	6	5	4	3	2	1	0
bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
Read/Write	R/W							
After reset	(all bits of the output latches are set to "1")							

Port 2 Function Register

P2FC (0009H)

	7	6	5	4	3	2	1	0
bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
Read/Write	W							
After reset	1	1	1	1	1	1	1	1
Function	0 : port 1 : address bus (A23 to A16)							

Read-modify-write is prohibited for registers P2FC.

Figure 3.5 (4) Port 2 Registers

3.5.3 Port 5 (P52 to P57)

Port 5 is a 6-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by the P5CR control register and the P5FR function register. A reset sets each output latch P57="0", P50, P52 to P56="1", and clears all P5CR control register and P5FR function register bits to "0". A reset sets P52-55 to input mode with pull-up resistors connected.

In addition to functioning as a general-purpose I/O port, port 5 also functions for CPU control/status signal I/O and for chip select signal output.

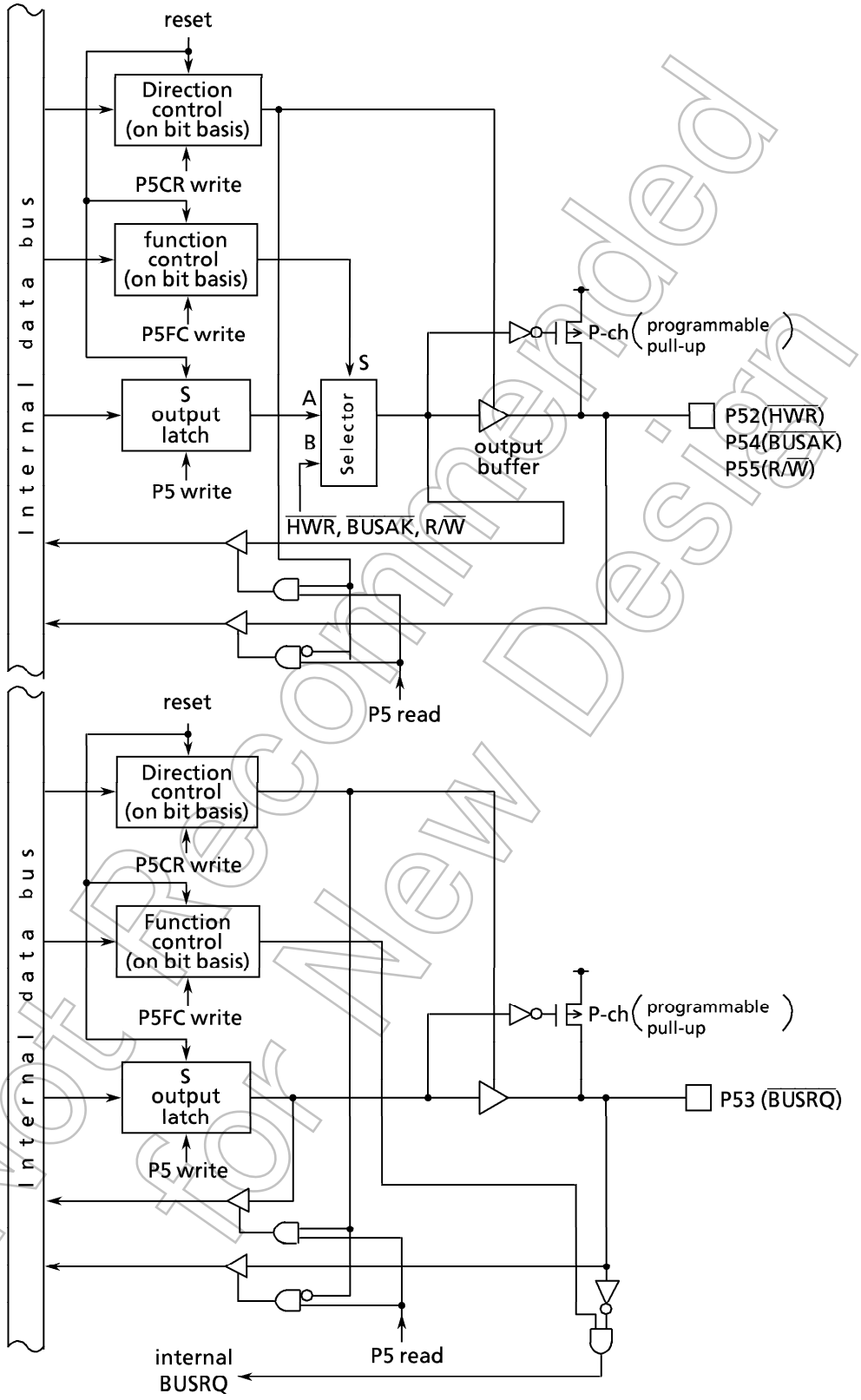


Figure 3.5 (5) Port 5 : P52 to P55

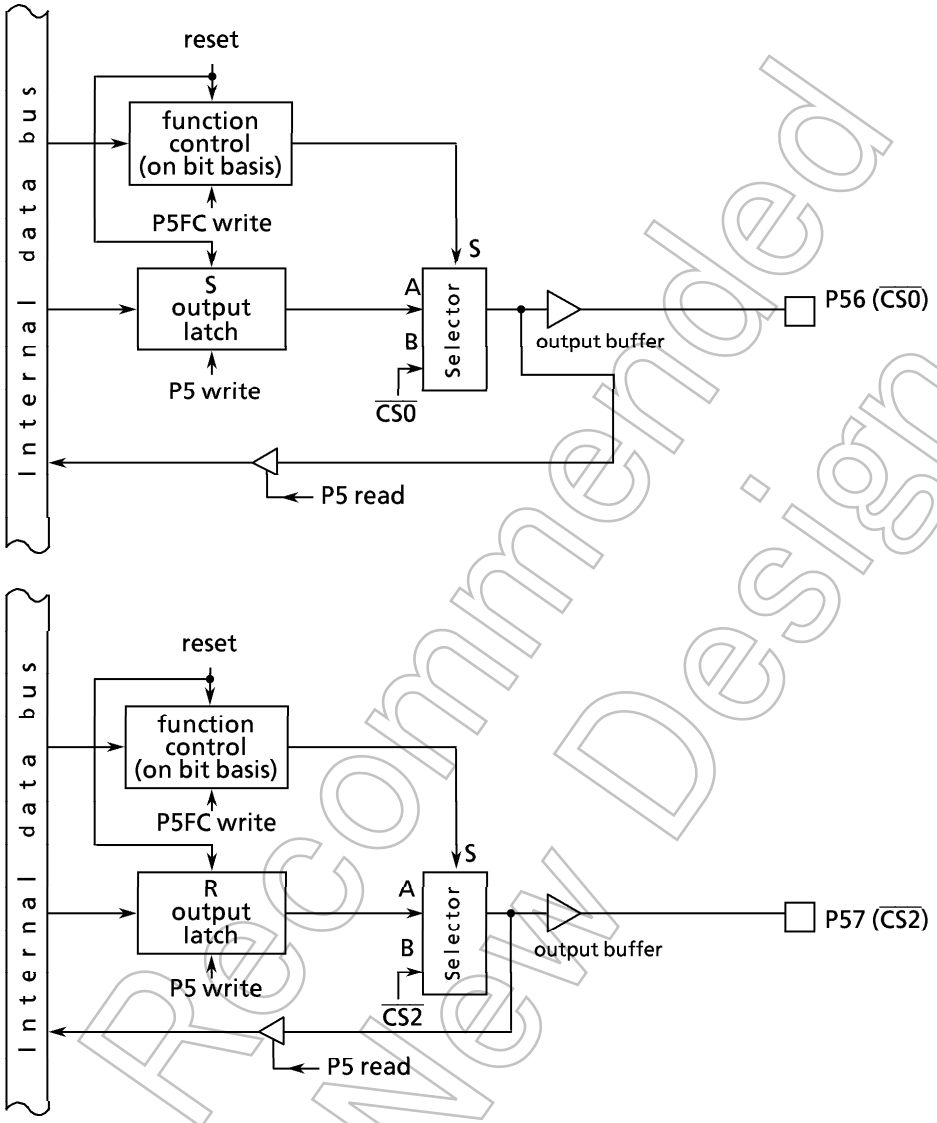


Figure 3.5 (6) Port 5 : P56, P57

Port 5 Register

P5 (000DH)

	7	6	5	4	3	2	1	0
bit Symbol	P57	P56	P55	P54	P53	P52		RDE
Read/Write	R/W							
After reset	output mode		input mode (with pull-up)					
	0	1	1	1	1	1		1

→ $\overline{RD}$ setting

0	PSRAM mode
1	—

Setting 0 also outputs $\overline{RD}$ when an internal area is being read.

Port 5 Control Register

P5CR (0010H)

	7	6	5	4	3	2	1	0
bit Symbol			P55C	P54C	P53C	P52C		
Read/Write	W							
After reset			0	0	0	0		
Function			0 : IN 1 : OUT					

→ Input/output setting

0	input
1	output

Port 5 Function Register

P5FC (0011H)

	7	6	5	4	3	2	1	0
bit Symbol	P57F	P56F	P55F	P54F	P53F	P52F		
Read/Write	W							
After reset	0	0	0	0	0	0		
Function	0 : PORT 1 : $\overline{CS2}$	0 : PORT 1 : $\overline{CS0}$	0 : PORT 1 : R/$\overline{W}$	0 : PORT 1 : $\overline{BUSAK}$	0 : PORT 1 : $\overline{BUSRQ}$	0 : PORT 1 : $\overline{HWR}$		

- Read-modify-write is prohibited for registers P5CR, P5FC.
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register P5.

→ $\overline{BUSRQ}$ setting

P5FC <P53F>	1
P5CR <P53C>	0

→ $\overline{BUSAK}$ setting

P5FC <P54F>	1
P5CR <P54C>	1

→ R/$\overline{W}$ setting

P5FC <P55F>	1
P5CR <P55C>	1

→ $\overline{HWR}$ setting

P5FC <P52F>	1
P5CR <P52C>	1

Figure 3.5 (7) Port 5 Registers

3.5.4 Port 6 (P60 to P67)

Port 6 is an 8-bit general-purpose output-only port. A reset sets all bits of the output latches of the port 6 register (P6) to "1".

In addition to functioning as a general-purpose I/O port, port 6 has the chip select signal output function ($\overline{CS1}$, $\overline{CS3}$) and DRAM control signal output function. The P6FC function register sets the port functions. A reset clears all P6FC register bits to "0" and sets P60 to P67 to general-purpose output port mode.

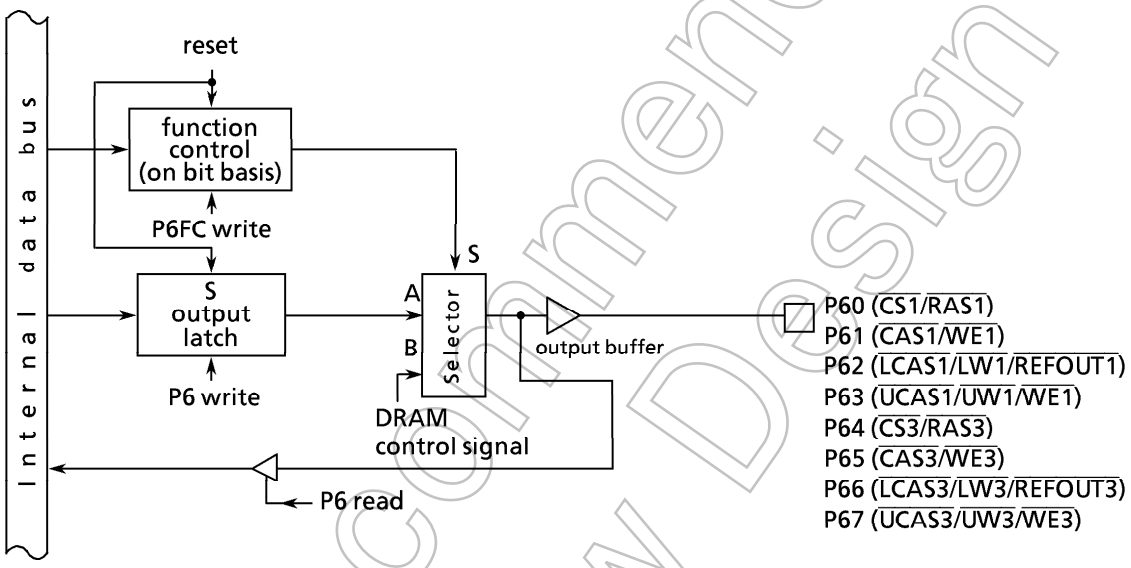


Figure 3.5 (8) Port 6

Port 6 Register

P6 (0012H)

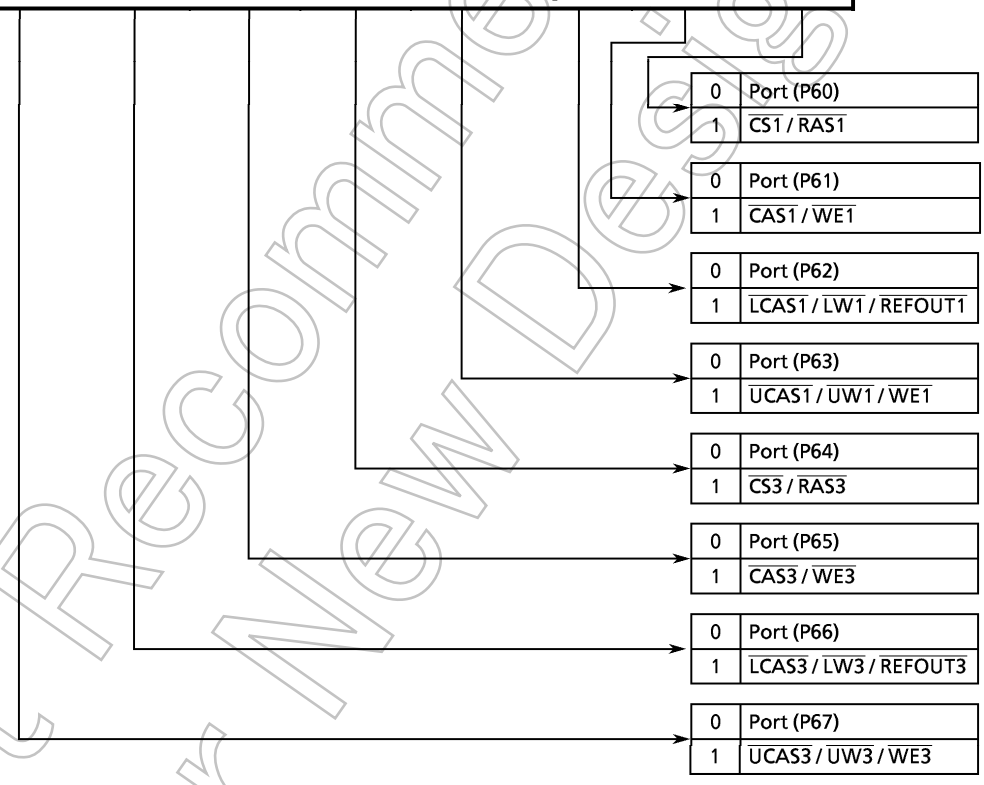
	7	6	5	4	3	2	1	0
bit Symbol	P67	P66	P65	P64	P63	P62	P61	P60
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Port 6 Function Register

P6FC (0015H)

	7	6	5	4	3	2	1	0
bit Symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : PORT 1 : DRAM control signal							

Read-modify-write is prohibited for registers P6FC



Note: The chip select/wait controller register (BnCS) selects pin functions of P60 ($\overline{CS1}$ / $\overline{RAS1}$) and P64 ($\overline{CS3}$ / $\overline{RAS3}$). The memory access method automatically determines the function of P61 to P63 and P65 to P67. (For details, see Table 3.7 (1) DRAM Pins.)

Figure 3.5 (9) Port 6 Registers

3.5.5 Port 7 (P70 to P77)

Port 7 is an 8-bit I/O port. Bits can be individually set as either inputs or outputs. A reset initializes the port as an input port with pull-up resistors, and sets all bits of the output latches in the port 7 register (P7) to "1". In addition to functioning as an I/O port, port 7 also has the pattern generator output function (PG0,1). PG0 is allocated to P70 to P73 and PG1 is allocated to P74 to P77. Writing "1" to the corresponding bits of the port 7 control register (P7CR) and function register (P7FC) enables PG output. A reset clears the function register (P7FC) to "0", and sets all bits to port mode.

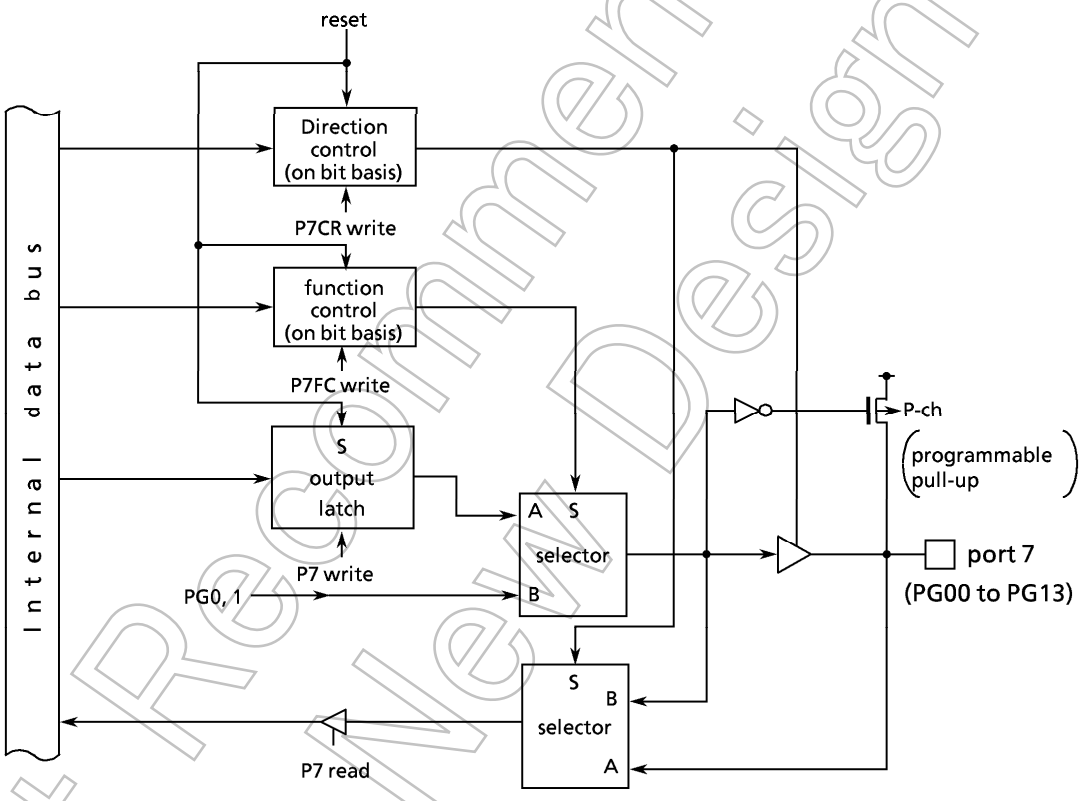


Figure 3.5 (10) Port 7

Port 7 Register

P7 (0013H)

	7	6	5	4	3	2	1	0
bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
Read/Write	R/W							
After reset	input mode (with pull-up)							
	1	1	1	1	1	1	1	1

Port 7 Control Register

P7CR (0016H)

	7	6	5	4	3	2	1	0
bit Symbol	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN 1 : OUT							

Port 7 input/output setting

0	input
1	output

Port 7 Function Register

P7FC (0017H)

	7	6	5	4	3	2	1	0
bit Symbol	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : PORT 1 : PG1-OUT				0 : PORT 1 : PG0-OUT			

Port 7 function setting

0	general-purpose port
1	stepping motor control / pattern generation port

- Read-modify-write is prohibited for registers P7CR, P7FC.
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register P7.

Figure 3.5 (11) Port 7 Registers

3.5.6 Port 8 (P80 to P87)

Port 8 is a general-purpose 8-bit I/O port. Bits can be individually set as either inputs or outputs. A reset initializes the port as an input port with pull-up resistors. In addition to functioning as I/O port pins, P80 can function for bus start (BS); and P81, for system clock output (SCOUT), P82 for WAIT input ($\overline{\text{WAIT}}$), P83 for non-maskable interrupt input ($\overline{\text{NMI2}}$), and P84-P87 for INT0-INT3 inputs. Writing "1" to the corresponding bits of the port 8 function register enables the bus start and system clock output functions. A reset clears the function register to "0" and sets all bits to port mode.

(1) Port 8: P80, P81, and P82 (BS, SCOUT, $\overline{\text{WAIT}}$)

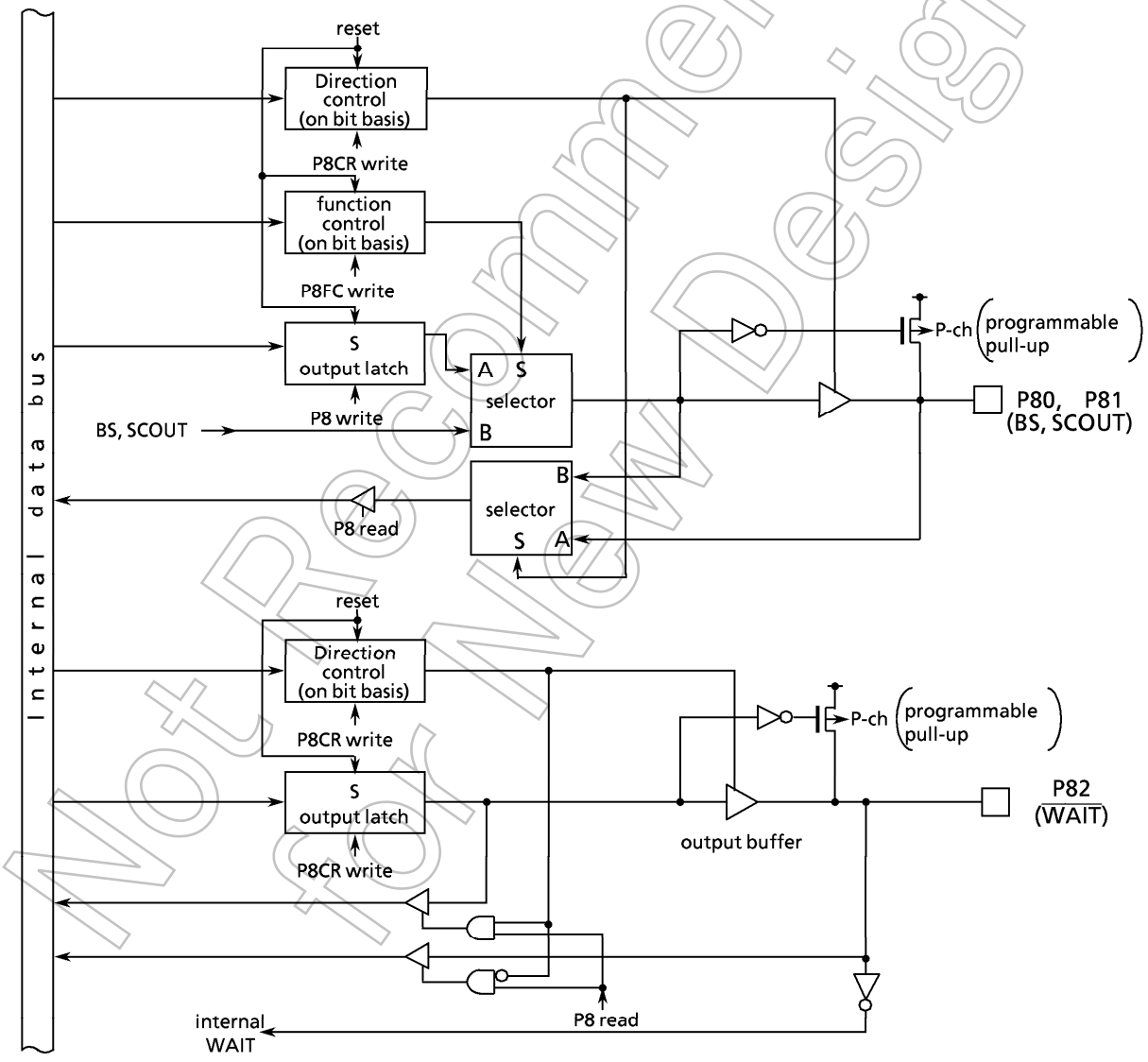


Figure 3.5 (12) Port 8 : P80, P81, and P82

(2) Port8 : P83 and P84 ($\overline{\text{NMI2}}$, INT0)

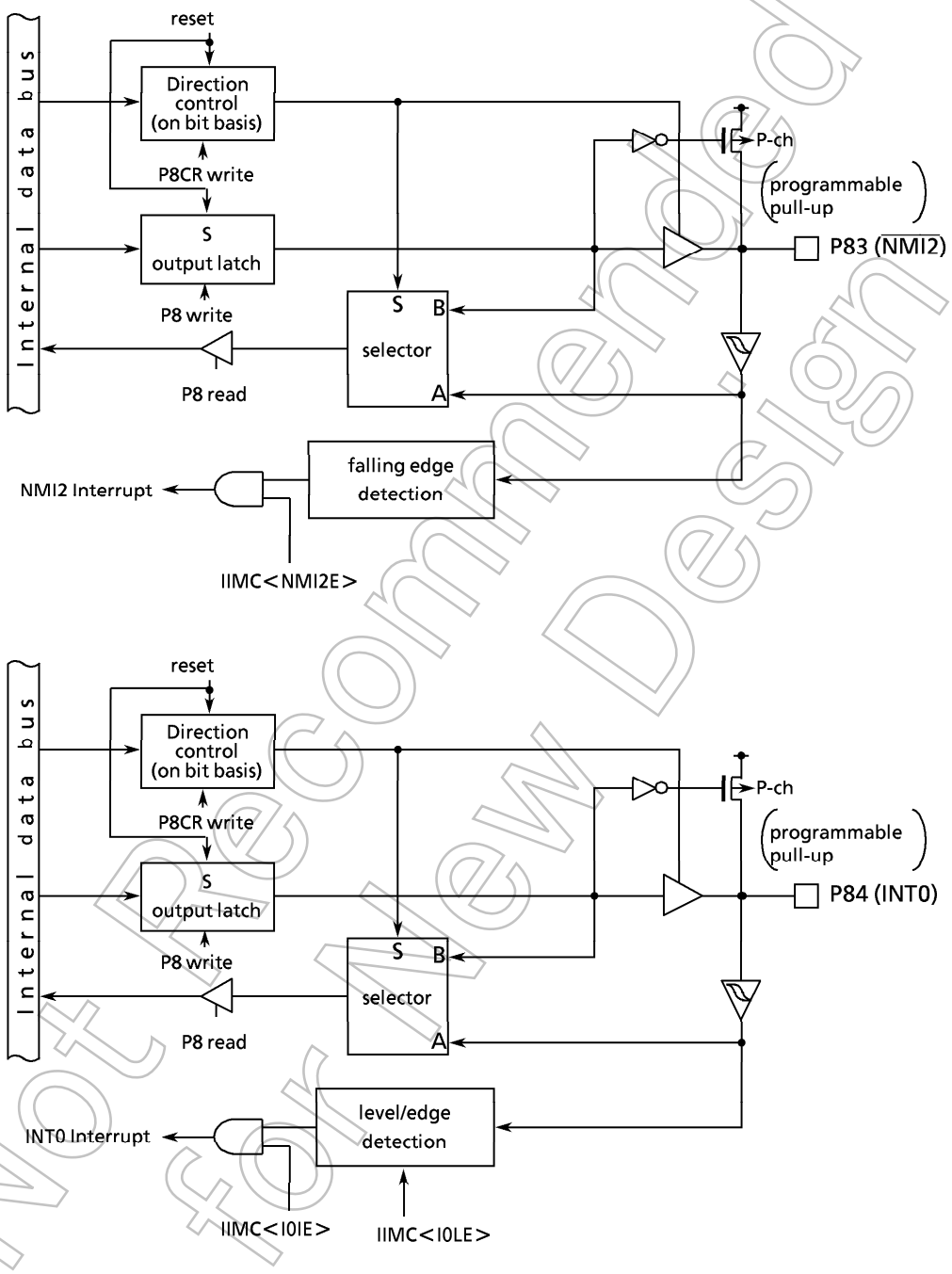


Figure 3.5 (13) Port 8 : P83 and P84

(3) Port 8 : P85, P86, and P87 (INT1, 2, 3)

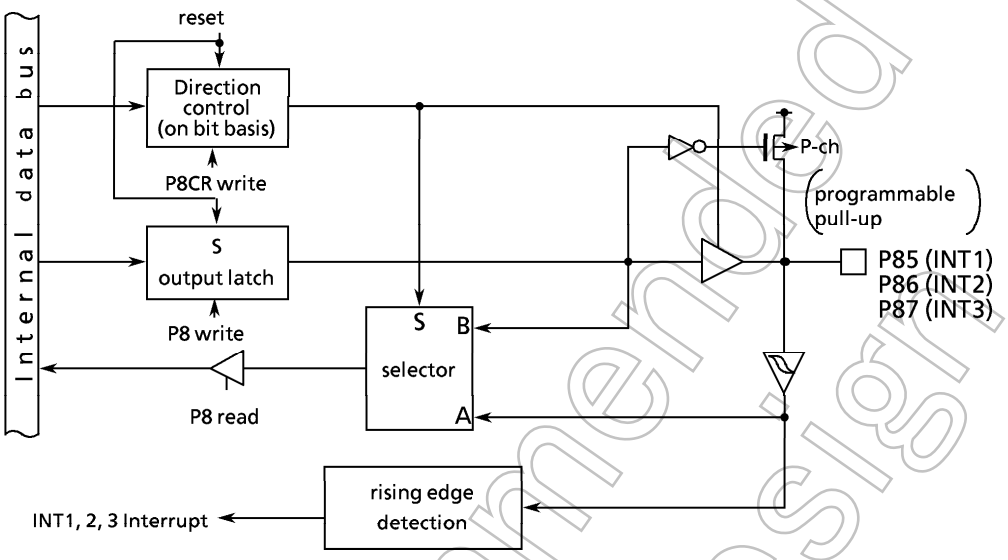


Figure 3.5 (14) Port 8 : P85, P86, and P87

Port 8 Register

P8 (0018H)

	7	6	5	4	3	2	1	0
bit Symbol	P87	P86	P85	P84	P83	P82	P81	P80
Read/Write	R/W							
After reset	input mode							
	1	1	1	1	1	1	1	1

Port 8 Control Register

P8CR (001AH)

	7	6	5	4	3	2	1	0
bit Symbol	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN				1 : OUT			

Port 8 input/output setting

0	input
1	output

Port 8 Function Register

P8FC (001BH)

	7	6	5	4	3	2	1	0
bit Symbol							P81F	P80F
Read/Write							W	
After reset							0	0
Function							0: PORT 1: SCOUT	0: PORT 1: BS

- Read-modify-write is prohibited for registers P8CR, P8FC.
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register P8.

0	Port (P80)
1	BS

0	Port (P81)
1	SCOUT

Note: When using P83 and P84 as the $\overline{\text{NMI2}}$ and INT0 pins respectively, set P8CR<P83C> and <P84C> to "0", and IIMC<NMI2E> and <I0IE> to "1".

Figure 3.5 (15) Port 8 Registers

3.5.7 Port 9 (P90 to P97)

Port 9 is a general-purpose 8-bit I/O port. Bits can be individually set as either inputs or outputs. A reset initializes the port as an input port with pull-up resistors. In addition to functioning as an I/O port, port 9 has the clock input and output pin functions for the 8-bit timers. Writing "1" to the corresponding bits of the port 9 function register (P9FC) enables the timer output functions. A reset clears the function register (P9FC) to "0" and sets all bits to port mode.

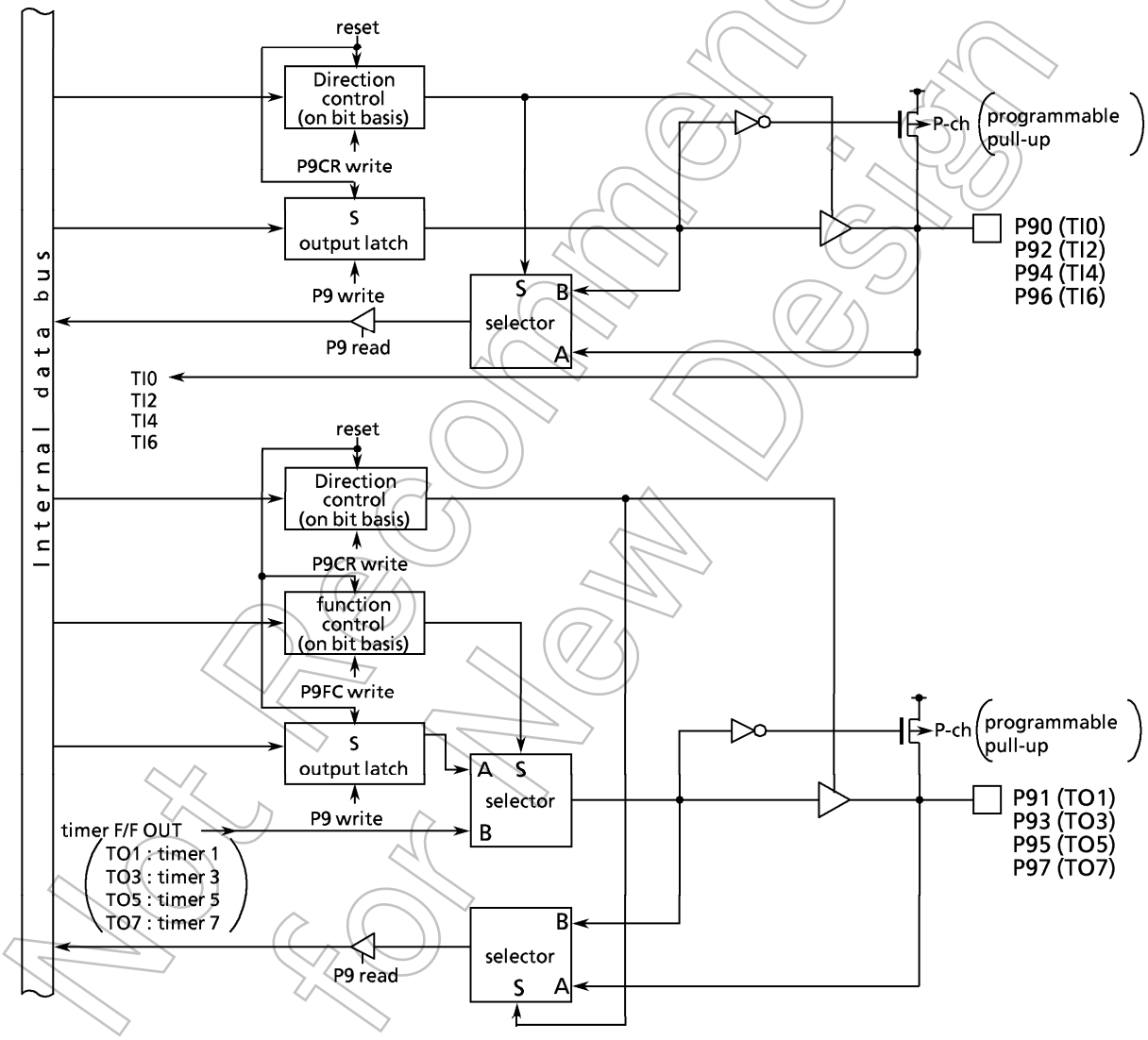


Figure 3.5 (16) Port 9

Port 9 Register

P9 (0019H)

	7	6	5	4	3	2	1	0
bit Symbol	P97	P96	P95	P94	P93	P92	P91	P90
Read/Write	R/W							
After reset	input mode (with pull-up)							
	1	1	1	1	1	1	1	1

Port 9 Control Register

P9CR (001CH)

	7	6	5	4	3	2	1	0
bit Symbol	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN 1 : OUT							

Port 9 input/output setting

0	input
1	output

Port 9 Function Register

P9FC (001DH)

	7	6	5	4	3	2	1	0
bit Symbol	P97F		P95F		P93F		P91F	
Read/Write			W		W		W	
After reset			0		0		0	
Function	0 : PORT 1 : TO7		0 : PORT 1 : TO5		0 : PORT 1 : TO3		0 : PORT 1 : TO1	

- Read-modify-write is prohibited for registers P9CR, P9FC.
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register P9.

P91 timer 1 output setting

P9FC <P91F>	1
P9CR <P91C>	1

P93 timer 3 output setting

P9FC <P93F>	1
P9CR <P93C>	1

P95 timer 5 output setting

P9FC <P95F>	1
P9CR <P95C>	1

P97 timer 7 output setting

P9FC <P97F>	1
P9CR <P97C>	1

Note: As no port/function switching register is supported for the P90/TI0, P92/TI2, P94/TI4, and P96/TI6 pins, even when the pins are used as input port pins, for example, data are input to the 8-bit timer.

Figure 3.5 (17) Port 9 Registers

3.5.8 Port A (PA0 to PA7)

Port A is a general-purpose 8-bit I/O port. Bits can be individually set as either inputs or outputs. A reset initializes the port as an input port with pull-up resistors, and sets all bits of the output latches in port A register (PA) to "1".

In addition to functioning as an I/O port, port A also functions as the I/O port for serial channels 0 and 1. Writing "1" to the corresponding bits of the port A function register enables this function. A reset clears the function register to "0" and sets all bits to port mode.

(1) Port A : PA0 and PA4 (TXD0/TXD1)

In addition to functioning as I/O port pins, PA0 and PA4 function as the TXD output pins for the serial channels.

PA0 and PA4 have the programmable open drain function.

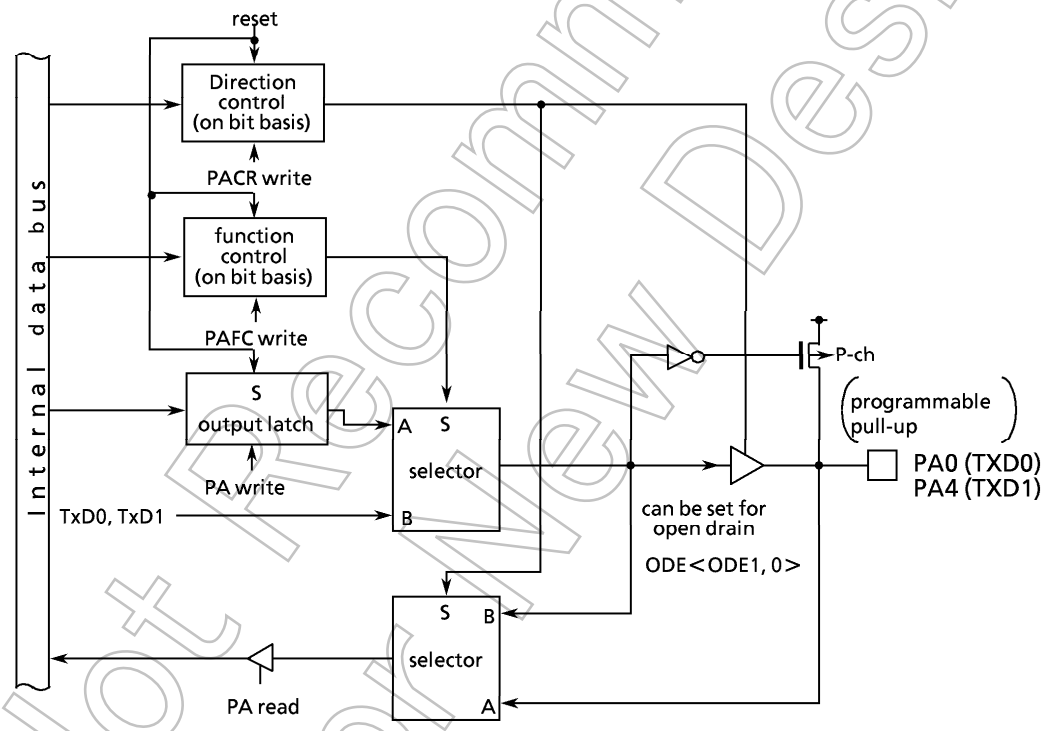


Figure 3.5 (18) Port A : PA0 and PA4

(2) Port A : PA1 and PA5 (RXD0,1)

In addition to functioning as I/O port pins, PA1 and PA5 function as the RXD input pins for the serial channels.

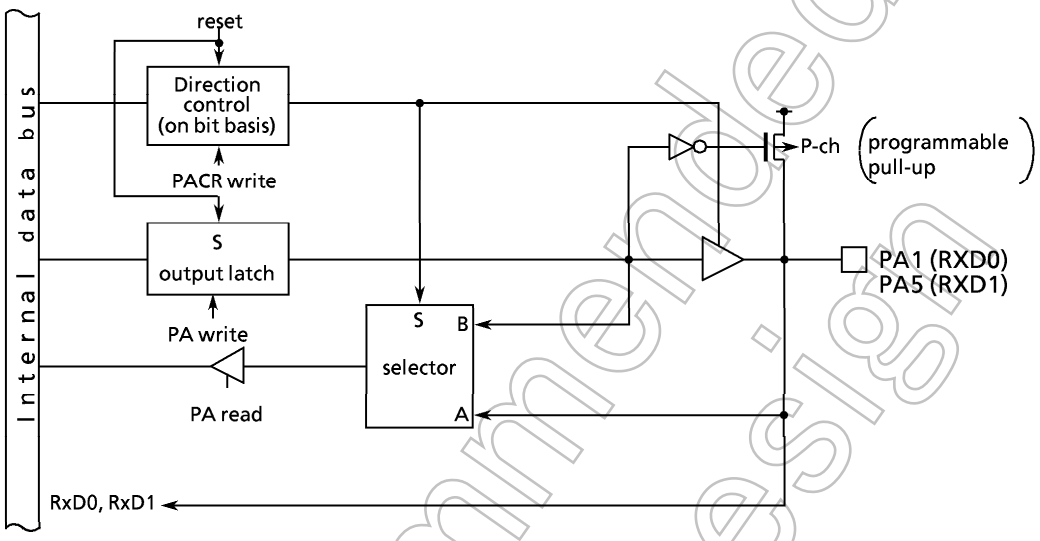


Figure 3.5 (19) Port A : PA1 and PA5

(3) Port A : PA2 and PA6 ($\overline{CTS0}$ / $\overline{CTS1}$)

In addition to functioning as I/O port pins, ports A2 and A6 function as the $\overline{CTS}$ input pins for the serial channels.

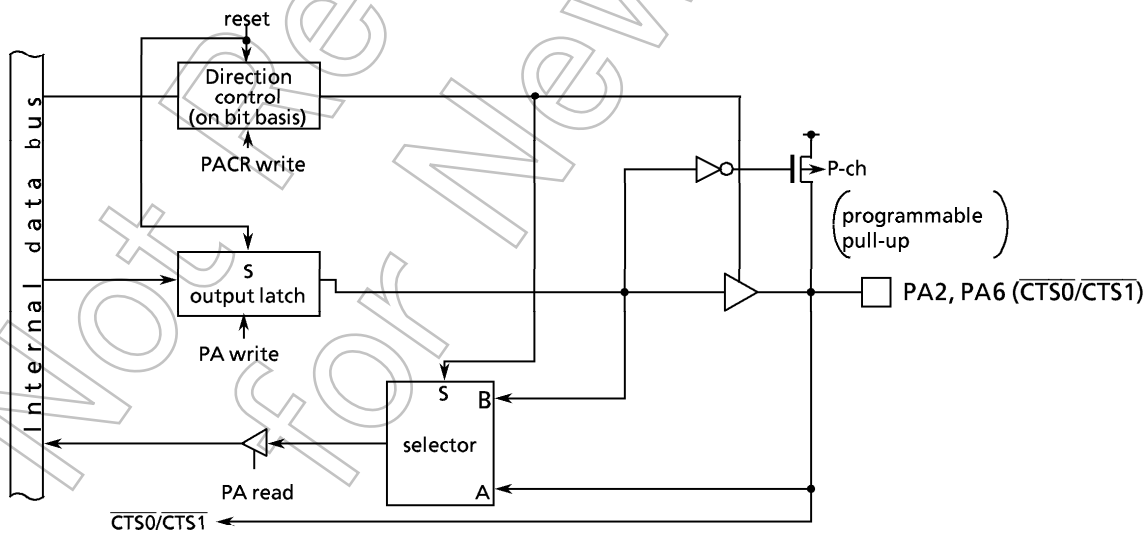


Figure 3.5 (20) Port A : PA2 and PA6

(4) Ports A3 and A7 (SCLK0/SCLK1)

In addition to functioning as general-purpose I/O port pins, PA3 and PA7 function as the SCLK I/O pins for the serial channels.

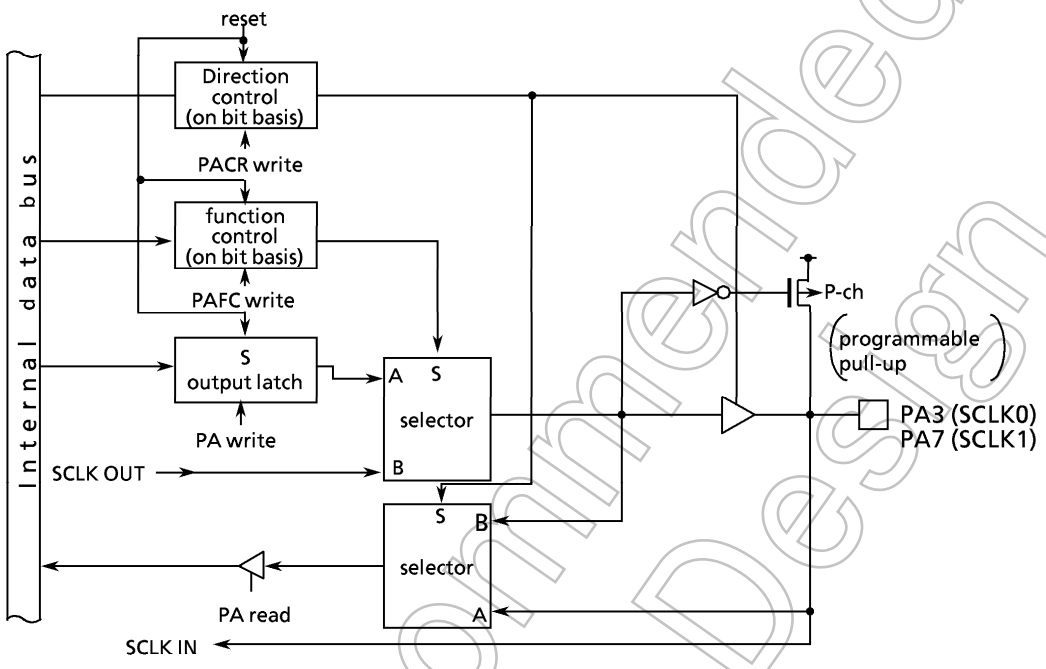


Figure 3.5 (21) Port A : PA3 and PA7

Port A Register

PA (001EH)

	7	6	5	4	3	2	1	0
bit Symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	R/W							
After reset	input mode (with pull-up)							
	1	1	1	1	1	1	1	1

Port A Control Register

PACR (0080H)

	7	6	5	4	3	2	1	0
bit Symbol	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN 1 : OUT							

Port A input/output setting

0	input
1	output

Port A Function Register

PAFC (0081H)

	7	6	5	4	3	2	1	0
bit Symbol	PA7F			PA4F	PA3F			PA0F
Read/Write	W			W	W			W
After reset	0			0	0			0
Function	0 : PORT 1 : SCLK1			0 : PORT 1 : TxD1	0 : PORT 1 : SCLK0			0 : PORT 1 : TxD0

- Read-modify-write is prohibited for registers PACR, PAFC.
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register PA.

PA0 TXD0 output setting (Note)

PAFC <PA0F>	1
PACR <PA0C>	1

PA3 SCLK0 output setting

PAFC <PA3F>	1
PACR <PA3C>	1

PA7 SCLK1 output setting

PAFC <PA7F>	1
PACR <PA7C>	1

PA4 TXD1 output setting (Note)

PAFC <PA4F>	1
PACR <PA4C>	1

Note : To set the TXD pins to open drain output, write "1" to bit 0 (for the TXD0 pin) or bit 1 (for the TXD1 pin) of the ODE register.
As no port/function switching register is supported for the PA1/RXD0 and PA5/RXD1 pins, even when the pins are used as input port pins, for example, data are input to SIO as a serial receive data.

Figure 3.5 (22) Port A Registers

3.5.9 Port B (PB0 to PB7)

Port B is a general-purpose 8-bit I/O port. Bits can be individually set as either inputs or outputs. A reset initializes the port as an input port with pull-up resistors, and sets all bits of the output latches in the port B register (PB) to "1". In addition to functioning as an I/O port, port B inputs the clock for 16-bit timers 8 and 9, and provides the output for 16-bit timer flip-flops 8, 9, A, and B. Writing "1" to the corresponding bits of the port B function register (PBFC) enables these functions. A reset clears the function register (PBFC) to "0", and sets all bits to port mode.

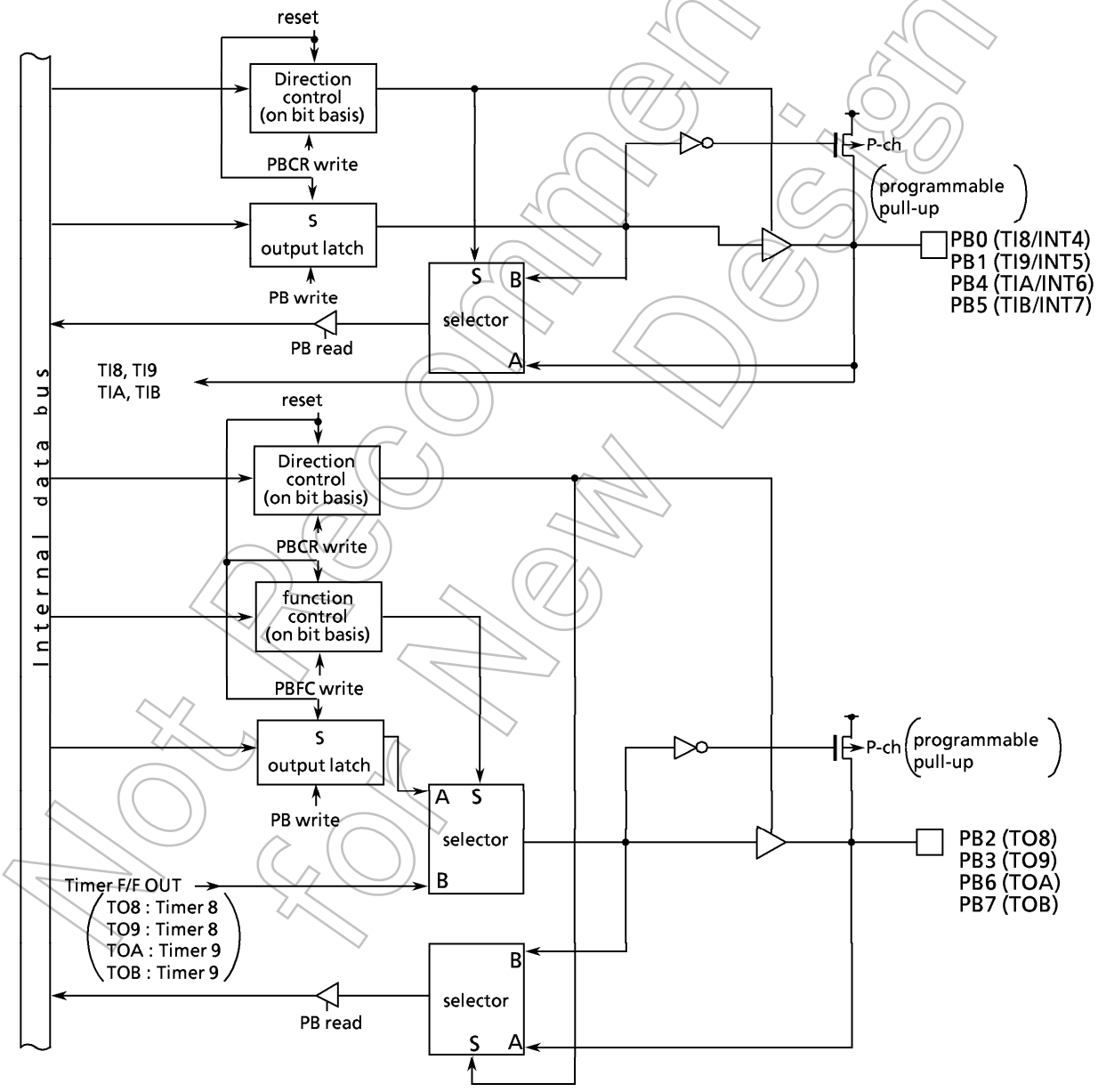


Figure 3.5 (23) Port B: PB0-PB7

Port B Register

PB (001FH)

	7	6	5	4	3	2	1	0
bit Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	R/W							
After reset	nput mode							
	1	1	1	1	1	1	1	1

Port B Control Register

PBCR (0082H)

	7	6	5	4	3	2	1	0
bit Symbol	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN				1 : OUT			

Port B input/output setting

0	input
1	output

Port B Function Register

PBFC (0083H)

	7	6	5	4	3	2	1	0
bit Symbol	PB7F	PB6F			PB3F	PB2F		
Read/Write	W	W			W	W		
After reset	0	0			0	0		
Function	0 : PORT 1 : TOB	0 : PORT 1 : TOA			0 : PORT 1 : TO9	0 : PORT 1 : TO8		

- Read-modify-write is prohibited for registers PBCR, PBFC.
- Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for register PB.

PB2 timer 8 output setting

PBFC <PB2F>	1
PBCR <PB2C>	1

PB3 timer 9 output setting

PBFC <PB3F>	1
PBCR <PB3C>	1

PB6 timer A output setting

PBFC <PB6F>	1
PBCR <PB6C>	1

PB7 timer B output setting

PBFC <PB7F>	1
PBCR <PB7C>	1

Note: As there is no port/function switching register for the PB0/TI8, PB1/T19, PB4/TIA, and PB5/TIB pins, even when the pins are used as input port pins, for example, data are input to the 16-bit timer.

Figure 3.5 (24) Port B Registers

3.5.10 Port C (PC0 to PC7)

Port C is an 8-bit input-only port that shares pins with the analog inputs.

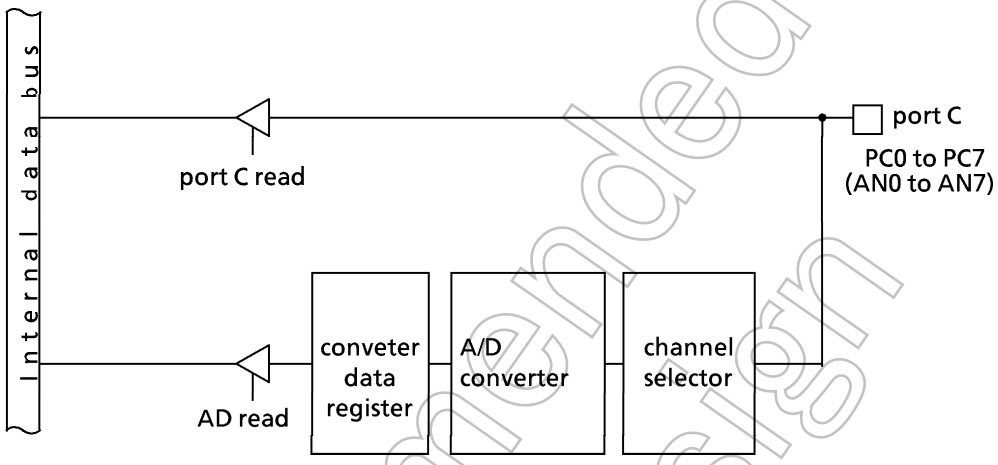


Figure 3.5 (25) Port C

Port C Register

	7	6	5	4	3	2	1	0
bit Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Read/Write	R							
After reset	input mode							

PC (0084H)

Note: Select the input channels for the A/D converter in A/D converter mode register ADMOD.

Figure 3.5 (26) Port C Registers

3.5.11 Port D (PD0 to PD4) and Port E (PE0 to PE7)

Ports D and E are respectively 5-bit and 8-bit general-purpose I/O ports. Bits can be individually set as either inputs or outputs. A reset initializes the ports as input ports with pull-up resistors. In addition to functioning as an I/O port, D0 also functions for the INT8 input.

(1) Port D : PD0 (INT8)

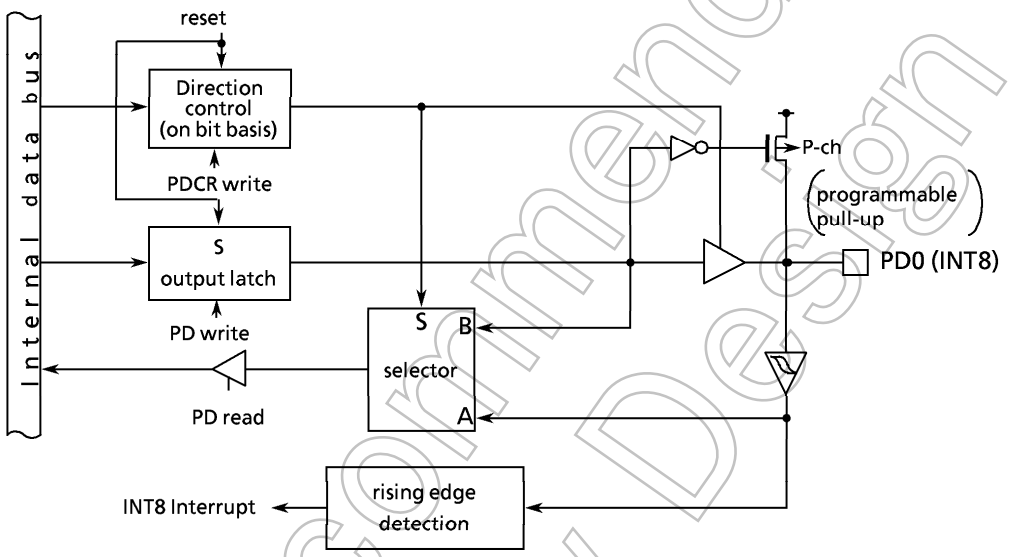


Figure 3.5 (27) Port D : PD0

(2) Ports D1 to D4 and E0 to E7

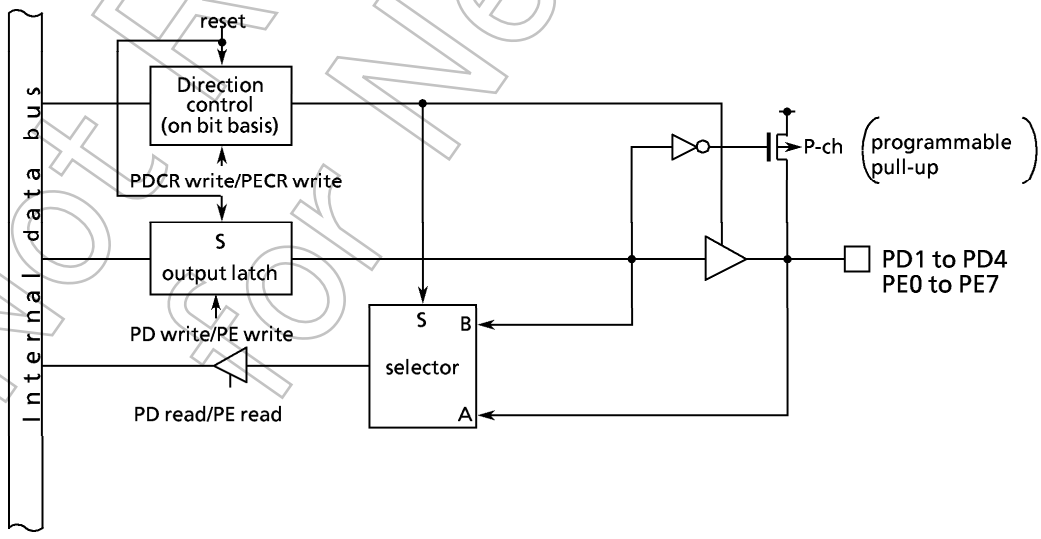


Figure 3.5 (28) Port D (PD1 to PD4) and Port E (PE0 to PE7)

Port D Register

PD (0085H)

	7	6	5	4	3	2	1	0
bit Symbol				PD4	PD3	PD2	PD1	PD0
Read/Write				R/W				
After reset				input mode				
				1	1	1	1	1

Port D Control Register

PDCR (0088H)

	7	6	5	4	3	2	1	0
bit Symbol				PD4C	PD3C	PD2C	PD1C	PD0C
Read/Write				W				
After reset				0	0	0	0	0
Function				0 : IN		1 : OUT		

Port D input/output setting

0	input
1	output

Port E Register

PE (008AH)

	7	6	5	4	3	2	1	0
bit Symbol	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Read/Write	R/W							
After reset	input mode							
	1	1	1	1	1	1	1	1

Port E Control Register

PECR (008CH)

	7	6	5	4	3	2	1	0
bit Symbol	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0 : IN				1 : OUT			

Port E input/output setting

0	input
1	output

・ Read-modify-write is prohibited for registers PDCR/ PECR.
・ Read-modify-write is prohibited for controlling ON/OFF of the pull-up resistor for registers PD/PE.

Figure 3.5 (29) Port D, E Registers

3.6 Chip Select/Wait Controller

TMP95C063 has an internal chip select/wait controller with a variable 4-block address area that controls the chip select ($\overline{CS0}$ to$\overline{CS3}$) and wait ($\overline{WAIT}$) pins, as well as the data bus size (8-bit or 16-bit).

TMP95C063 also has an external data bus size selection pin (AM8/$\overline{16}$). (See 3.1.2, External Data Bus Selection Pin.)

3.6.1 Control Registers

Table 3.6 (1) shows the control registers.

A block address area is controlled by the CS/wait control register (B0CS, B1CS, B2CS, B3CS, BEXCS), the memory start address register, and memory address mask register. (See 3.6.2, Specifying Address Areas.)

Table 3.6 (1) Chip Select/Wait Control Register

		7	6	5	4	3	2	1	0
B0CS (0090H) (No RMW)	bit Symbol	B0E	—	B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
	Read/Write	W	—	W		W	W		
	After reset	0	—	0	0	0	0	0	0
	Function	0: disable 1: enable	—	00: ROM/SRAM 01: PSRAM 10: Don't Care 11: Don't Care		0: 16 BIT 1: 8 BIT	000: 2 WAIT 001: 1 WAIT 010: 1 WAIT + N 011: 0 WAIT	100: N WAIT 101: — 110: — 111: —	
B1CS (0091H) (No RMW)	bit Symbol	B1E	—	B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
	Read/Write	W	—	W		W	W		
	After reset	0	—	0	0	0	0	0	0
	Function	0: disable 1: enable	—	00: ROM/SRAM 01: PSRAM 10: DRAM 11: Don't Care		0: 16 BIT 1: 8 BIT	000: 2 WAIT 001: 1 WAIT 010: 1 WAIT + N 011: 0 WAIT	100: N WAIT 101: — 110: — 111: —	
B2CS (0092H) (No RMW)	bit Symbol	B2E	B2M	B2OM2	B2OM1	B2BUS	B2W2	B2W1	B2W0
	Read/Write	W	W	W		W	W		
	After reset	1	0	0	0	0	0	0	0
	Function	0: disable 1: enable	0: 16M byte area 1: Sets area.	00: ROM/SRAM 01: PSRAM 10: Don't Care 11: Don't Care		0:16 BIT 1: 8 BIT	000: 2 WAIT 001: 1 WAIT 010: 1 WAIT + N 011: 0 WAIT	100: N WAIT 101: — 110: — 111: —	
B3CS (0093H) (No RMW)	bit Symbol	B3E	—	B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
	Read/Write	W	—	W		W	W		
	After reset	0	—	0	0	0	0	0	0
	Function	0: disable 1: enable	—	00: ROM/SRAM 01: PSRAM 10: DRAM 11: Don't Care		0: 16 BIT 1: 8 BIT	000: 2 WAIT 001: 1 WAIT 010: 1 WAIT + N 011: 0 WAIT	100: N WAIT 101: — 110: — 111: —	
BEXCS (008FH) (No RMW)	bit Symbol	—	—	—	—	BEXBUS	BEXW2	BEXW1	BEXW0
	Read/Write	—	—	—	—	W	W		
	After reset	—	—	—	—	0	0	0	0
	Function	—	—	—	—	0: 16 BIT 1: 8 BIT	000: 2 WAIT 001: 1 WAIT 010: 1 WAIT + N 011: 0 WAIT	100: N WAIT 101: — 110: — 111: —	

Note : Read-modify-write is prohibited for registers B0CS, B1CS, B2CS, B3CS and BEXCS.

(1) Master Enable

Bit 7 of the control register (B0E, B1E, B2E, B3E) is the master enable/disable bit. Set the bit to "0" to disable, or to "1" to enable the setting. A reset sets B0E, B1E, and B3E to "0" (disabled), and sets B2E to "1" (enabled).

(2) Data Bus Size Selection

Bit 3 of the control registers (B0BUS, B1BUS, B2BUS, B3BUS, and BEXBUS) specifies the width of the data bus. Set "0" to access memory in 16-bit data bus mode, or to "1" in 8-bit data bus mode. Note that this bit is valid only in 16-bit bus mode (when the AM8/$\overline{16}$ pin is "0"). In 8-bit bus mode (when the AM8/$\overline{16}$ pin is "1"), memory access to all address areas uses 8-bit data bus mode, regardless of the value of bit 3. (See 3.1.2, External Data Bus Size Selection Pin.)

Note that when using CS1 and CS3 as DRAM, even when the AM8/$\overline{16}$ pin is "1", B1BUS and B3BUS must be set to "1".

This way of changing the data bus size depending on the address being accessed is called "dynamic bus sizing". See 3.6 (2) for details of this bus operation.

Table 3.6 (2) Dynamic Bus Cycling

Operand Data Width	Operand Start Address	Memory Data Width	CPU Address	CPU Data	
				D15 to D8	D7 to D0
8 bits	2n + 0 (even-numbered)	8 bits	2n + 0	xxxxx	b7 to b0
		16 bits	2n + 0	xxxxx	b7 to b0
	2n + 1 (odd-numbered)	8 bits	2n + 1	xxxxx	b7 to b0
		16 bits	2n + 1	b7 to b0	xxxxx
16 bits	2n + 0 (even-numbered)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (odd-numbered)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	xxxxx	b15 to b8
32 bits	2n + 0 (even-numbered)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
			2n + 2	xxxxx	b23 to b16
			2n + 3	xxxxx	b31 to b24
		16 bits	2n + 0	b15 to b8	b7 to b0
			2n + 2	b31 to b24	b23 to b16
	2n + 1 (odd-numbered)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
			2n + 3	xxxxx	b23 to b16
			2n + 4	xxxxx	b31 to b24
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	b23 to b16	b15 to b8
			2n + 4	xxxxx	b31 to b24

xxxxx : During a read, indicates that bus input data are ignored; during a write, indicates that the bus is set to high impedance and that the bus write strobe signal remains inactive.

(3) Wait Control

Bits 2, 1, and 0 of the control register (B0W2·1·0, B1W2·1·0, B2W2·1·0, B3W2·1·0, BEXW2·1·0) specify the number of waits. Setting these bits to "000" inserts a wait of two states, irrespective of the $\overline{WAIT}$ pin state. Setting these bits to "001" inserts a wait of one state, irrespective of the $\overline{WAIT}$ pin state. Setting to "010" samples the state of the $\overline{WAIT}$ pin after inserting a wait of one state. If the $\overline{WAIT}$ pin is low, the wait continues and the bus cycle is extended until the pin goes high. Setting to "011" ends the bus cycle without a wait, regardless of the $\overline{WAIT}$ pin state. Setting to "100" continuously samples the $\overline{WAIT}$ pin state and inserts a wait if the pin is low, extending the bus cycle until the pin goes high. Figure 3.6 (1), (2), (3) shows the timing setting to 0+N wait in the case of N=1, 2, 3. The $\overline{WAIT}$ pin status is sampled at the falling edge of SCOUT. See "Chapter3 TLCS-900 CPU Figure 7 (1) to (5)" setting<BxW2 · 1 · 0>except 0+N wait. A reset clears these bits to "000" (two-wait mode).

Note : If a contention between DRAM access and refreshing occurs when using DRAM, the length of the refresh cycle is added to the set number of wait states.

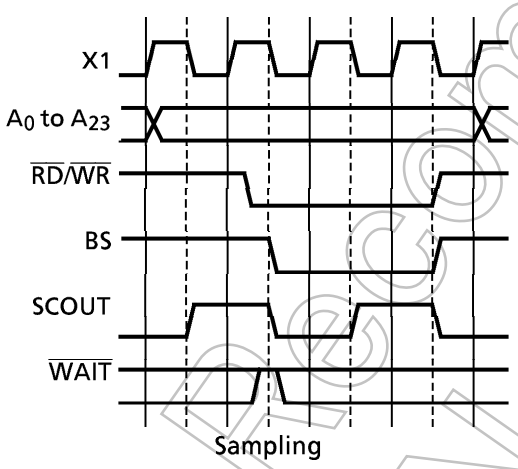


Figure 3.6 (1) 0 + N WAIT Read / Write Cycle (N = 0)

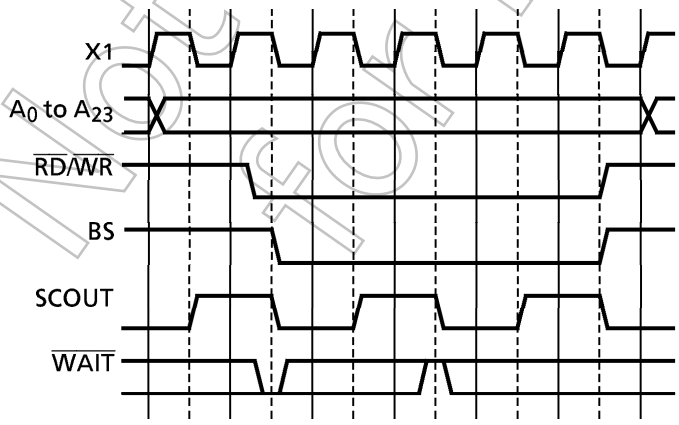


Figure 3.6 (2) 0 + N WAIT Read / Write Cycle (N = 1)

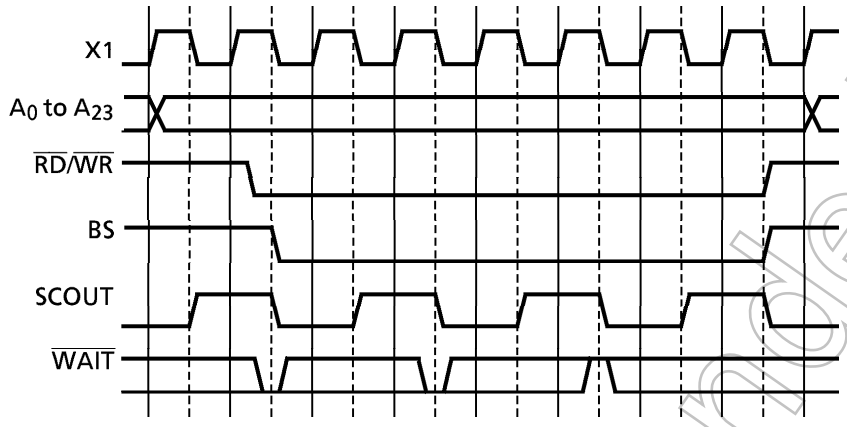


Figure 3.6 (3) 0 + N WAIT Read / Write Cycle (N = 2)

(4) ROM/PSRAM/DRAM Selection

Bits 5 and 4 of the control register (B0OM1·0, B1OM1·0, B2OM1·0, B3OM1·0) select the memory to be accessed. Figure 3.6 (4) shows the waveforms for ROM mode and PSRAM mode. (For details of DRAM mode, see 3.7, DRAM Controller.)
Setting these bits to "00" accesses memory by outputting the ROM mode chip select waveforms; setting them to "01" accesses memory by outputting the PSRAM mode chip select waveforms.

Setting to "10" (CS1 and CS3 only), together with the DRAM controller settings, accesses memory by outputting the RAS waveforms for DRAM. A reset clears the bits to "00".

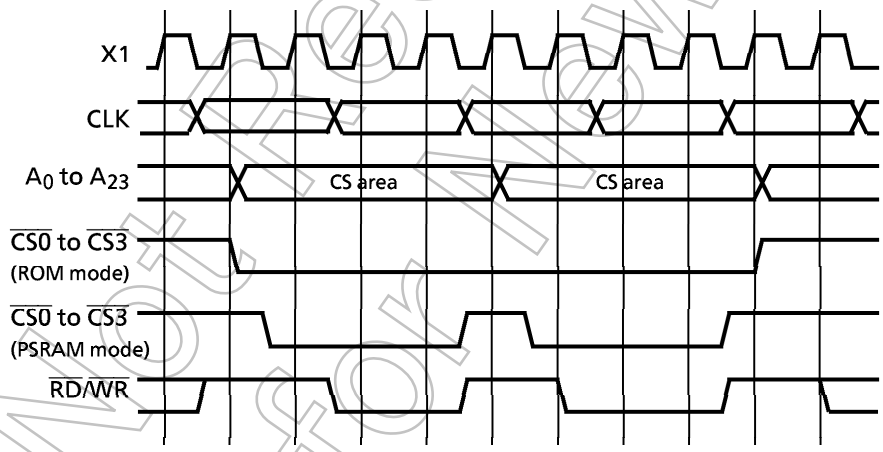


Figure 3.6 (4) Chip Select ($\overline{CS0}$ to $\overline{CS3}$) Timing (ROM Mode and PSRAM Mode)

(5) Bus Size and Wait Control Outside $\overline{CS0}$ to $\overline{CS3}$ Area

The BEXCS register controls the bus size and wait when locations outside the $\overline{CS0}$ to $\overline{CS3}$ address areas are accessed. This register contains no master enable bit; the register settings are always enabled for access to areas outside $\overline{CS0}$ to $\overline{CS3}$. The bit meanings are the same as for $\overline{CS0}$ to $\overline{CS3}$.

(6) Accessing 16M-byte Area/Address Setting Area

Setting B2CS<B2M> to "0" selects CS2 with a 16M-byte address area (000080H to FFFFFFH). Setting B2CS<B2M> to "1" selects CS2 with the address area specified by memory start address register MSAR2 and memory address mask register MAMR2, as in the case of CS0 and CS1. A reset clears this bit to "0".

3.6.2 Address area specification

An address area is specified by the corresponding memory start address register (MSAR0, MSAR1, MSAR2, MSAR3) and address mask register (MAMR0, MAMR1, MAMR2, MAMR3). At each bus cycle, the chip select controller compares the address on the bus with the value in the memory start address register. When the address is compared, the value of the address mask register specifies which bits of the comparison result to ignore. If the result of the comparison is a match, this indicates an access to the specified area. If the block is enabled (B0E-B3E = "1"), the corresponding chip select pin ($\overline{CS0}$ to $\overline{CS3}$) outputs a low strobe signal. If address area settings overlap, the block with the smallest CS number is selected. This applies even if CS2 is set to 16 Mbytes.

When the set address area overlaps with the internal I/O area, the functions as the internal I/O area take priority of the set address area.

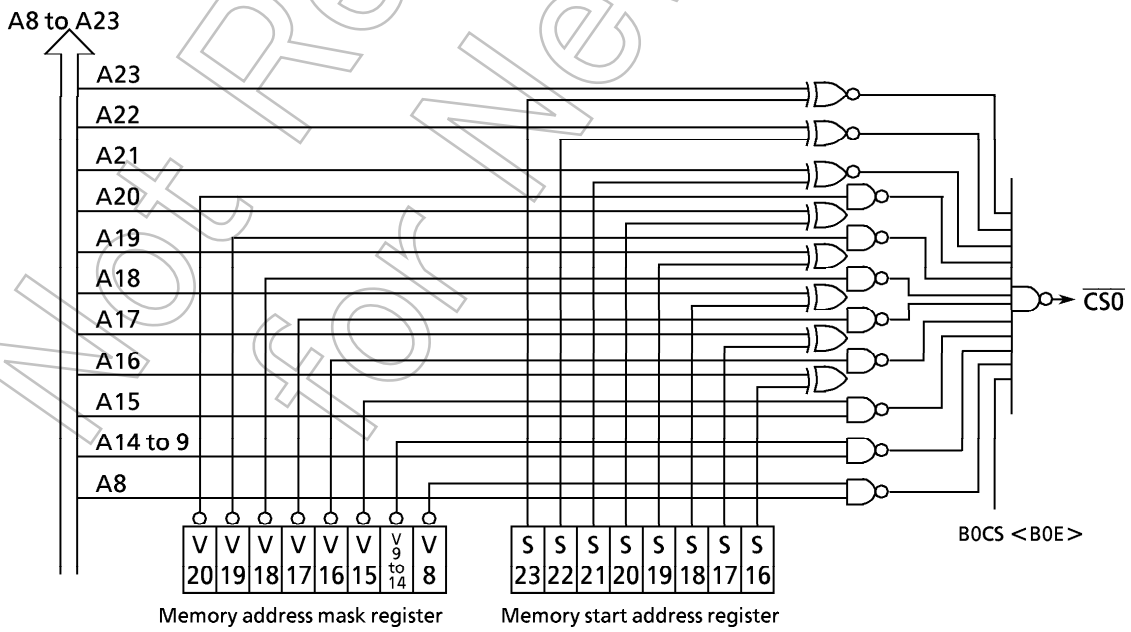


Figure 3.6 (5) $\overline{CS0}$ Address Decoder Block Diagram

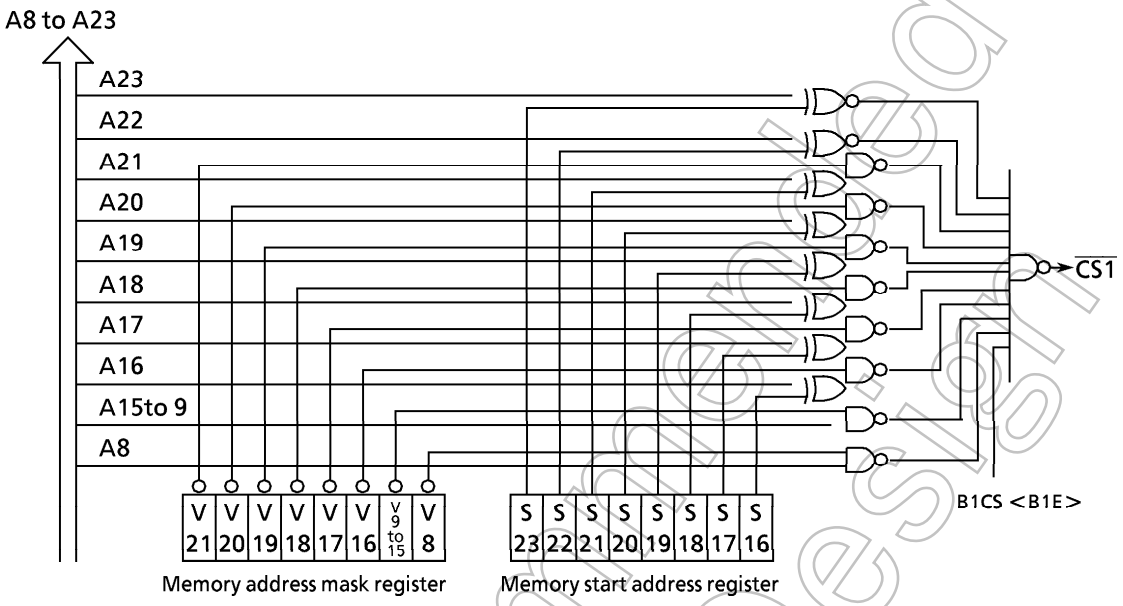


Figure 3.6 (6) $\overline{CS1}$ Address Decoder Block Diagram

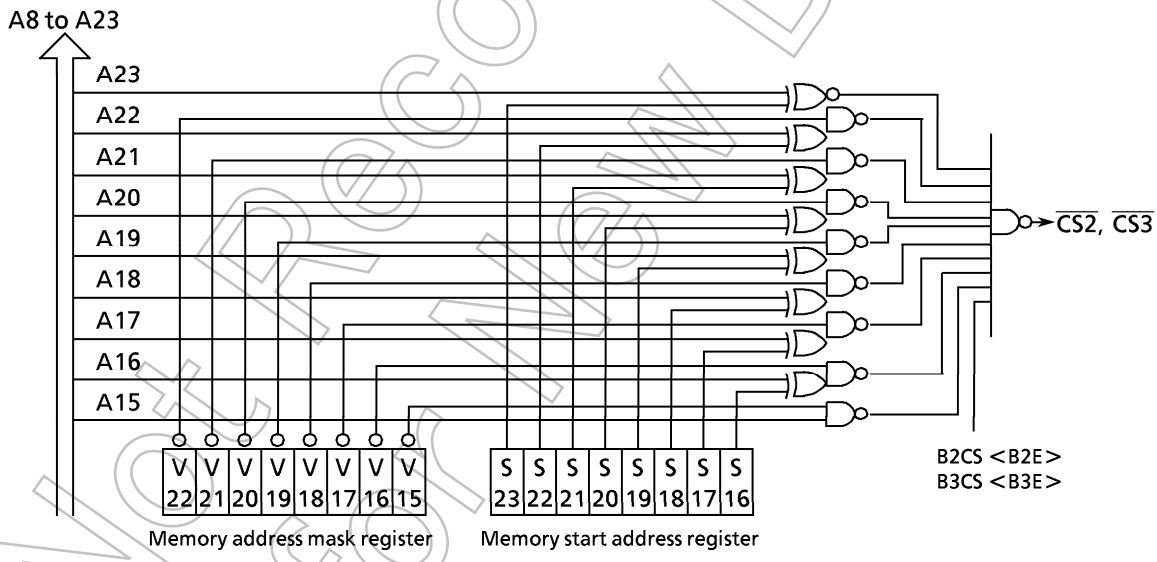


Figure 3.6 (7) $\overline{CS2}$ and $\overline{CS3}$ Address Decoder Block Diagram

(1) Memory Start Address Registers and Memory Address Mask Registers

Memory address register ($\overline{CS0}$ to $\overline{CS3}$)

MSAR0 (0094H) / MSAR1 (0096H) / MSAR2 (0098H) / MSAR3 (009AH)	7	6	5	4	3	2	1	0
bit Symbol	S23	S22	S21	S20	S19	S18	S17	S16
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets start addresses for A23 to A16.							

→ Sets $\overline{CS0}$ to $\overline{CS3}$ start addresses.

Table 3.6 (3) Memory Start Address Registers

Memory address mask register ($\overline{CS0}$)

MAMR0 (0095H)	7	6	5	4	3	2	1	0
bit Symbol	V20	V19	V18	V17	V16	V15	V14to 9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	0: enable compare 1: mask compare							

→ Controls data comparison with $\overline{CS0}$ addresses A8 to A20.

Memory address mask register ($\overline{CS1}$)

MAMR1 (0097H)	7	6	5	4	3	2	1	0
bit Symbol	V21	V20	V19	V18	V17	V16	V15to 9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	0: enable compare 1: mask compare							

→ Controls data comparison with $\overline{CS1}$ addresses A8 to A20.

Memory address mask registers ($\overline{CS2}$ and $\overline{CS3}$)

MAMR2 (0099H) / MAMR3 (009BH)	7	6	5	4	3	2	1	0
bit Symbol	V22	V21	V20	V19	V18	V17	V16	V15
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	0: enable compare 1: mask compare							

→ Controls data comparison with $\overline{CS2}$, $\overline{CS3}$ addresses A15 to A22.

Table 3.6 (4) Memory Address Mask Registers

MSAR0 to MSAR3<S23> to <S16> correspond to addresses A23 to A16. S15, S14 to S9, and S8, which correspond to A15, A14 to A9, and A8 respectively, default to "0". MAMR0<V20> to <V8> enable or mask the comparison of the MSAR0 value and the corresponding address. <V20> to <V8> correspond to <S20> to <S16>, S15, S14 to S9, and S8. V21, V22, and V23, which correspond to <S21>, <S22>, and <S23>, default to "0", meaning that comparison is always enabled for these bits.

"Comparison enabled/masked" means that (for the $\overline{CS0}$ registers MSAR0 and MAMR0, for example):

If <V16> is set to "1" to mask comparison, then:

The comparison between the <S16> value and address A16 is masked and the <S16> value is ignored.

If <V16> is set to "0" to enable comparison, then:

The comparison between the <S16> value and address A16 is enabled. Only if the <S16> value and A16 value match, $\overline{CS0}$ is enabled.

The same procedure applies for $\overline{CS1}$, $\overline{CS2}$, and $\overline{CS3}$.

After a reset, MSAR0, MSAR1, MSAR2, and MSAR3 are set to "FFH"; MAMR0, MAMR1, MAMR2, and MAMR3 are set to "FFH"; bits B0E, B1E, and B3E of the control register are reset to "0" to disable $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS3}$; B2M is cleared to "0"; B2E is set to "1"; and $\overline{CS2}$ is enabled for the 000080H to FFFFFFH (16M-byte) address area.

(2) Setting Start Addresses

The address decoder outputs $\overline{CS}$ based on the specified start address and area size.

Since only the upper 8 bits of the start address (A16 to A23) are decoded, as shown in the block diagram below, the start address is set in 64 K-byte steps.

That is, the DRAM start address is set to any 64 K-byte boundary, starting from 000000H.

However, note that the start address changes depending on the MAMR setting.

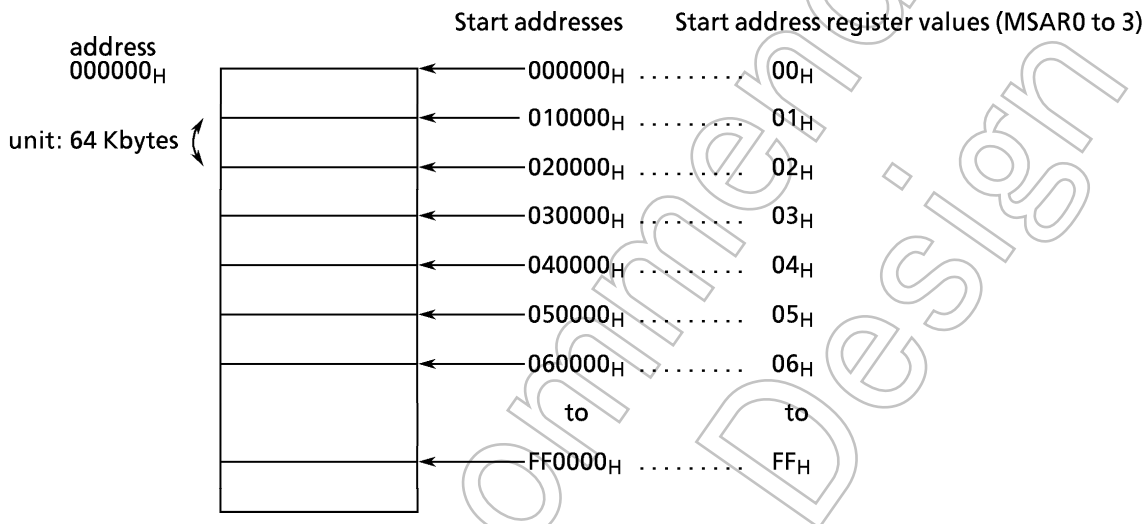


Figure 3.6 (8) Start Address Settings

(3) Setting Address Areas

The address areas are specified by setting the memory address mask registers (MAMR0 to 3).

As shown in the address decoder block diagrams (Figure 3.6 (2) to (4)), the address areas for which the chip select signals are output can be specified by enabling or masking comparison of the A8 to A20 (for $\overline{CS0}$), A8 to A21 (for $\overline{CS1}$), and A15 to A22 (for $\overline{CS2}$ and $\overline{CS3}$).

size / CS	256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
CS0	○	○	○	○	○	○	○	○	○		
CS1	○	○		○	○	○	○	○	○	○	
CS2			○	○	○	○	○	○	○	○	○
CS3			○	○	○	○	○	○	○	○	○

Table 3.6 (5) Chip Select and Area Size

(4) Setting Procedure

① Set the memory start address register (MSAR).
Set the area start address.

② Set the memory address mask register (MAMR).
Set the area.

③ Set the control register (BnCS).
Set the bus size, number of waits, and area enable/disable.

Example :

Set the $\overline{CS0}$ area as 010000_H to $01FFFF_H$ (64 Kbytes), a 16-bit bus, and zero waits:

MSAR0 = 01_H ..... start address 010000_H
MAMR0 = 07_H ..... address area 64 Kbytes
B0CS = 83_H ..... 16-bit bus, zero waits, $\overline{CS0}$ enabled, ROM mode access

3.7 Dynamic RAM (DRAM) Controller

TMP95C063 incorporates a 2-channel DRAM controller for interface with × 8/16-bit DRAM. The DRAM controller consists of a control circuit to refresh the DRAM , an access circuit for reading and writing , and a row/ column address multiplexer.

1) Refresh mode
   $\overline{CAS}$ before $\overline{RAS}$ refreshing

2) Refresh interval
   Programmable (31-195 states)

3) Refresh cycle width
   Programmable (2-9 states)

4) Two mapping areas
   $\overline{CS1}$ and $\overline{CS3}$

5) Address mapping size
   $\overline{CS1}$ area : 256-4 Mbytes
   $\overline{CS3}$ area : 32 Kbytes-8 Mbytes

6) Memory access mode
   2$\overline{CAS}$ / 2$\overline{WE}$ selectable

7) Memory access address length
   8-11 bits selectable

8) Wait control
   In accordance with CS/WAIT controller setting

9) Arbitration of refresh/access contention
   Refresh has higher priority. Wait states are automatically inserted in the access cycle.

Control Register

DREFCR1 (009CH)

	7	6	5	4	3	2	1	0
bit Symbol	DMI1	RS12	RS11	RS10	RW12	RW11	RW10	RC1
Read/Write	R / W							
After reset	0	0	0	0	0	0	0	0
Function	dummy cycle 0: no disable 1: dummy cycle	refresh cycle insertion interval 000: 31 states 001: 62 states 010: 78 states 011: 97 states 100: 109 states 101: 124 states 110: 154 states 111: 195 states			refresh cycle width 000: 2 states 001: 3 states 010: 4 states 011: 5 states 100: 6 states 101: 7 states 110: 8 states 111: 9 states			refresh cycle 0 : no refresh cycle 1 : refresh cycle

Refresh cycle control

0	no refresh cycle
1	refresh cycle

Refresh cycle width control

RW12	RW11	RW10	Refresh cycle width
0	0	0	2 states
0	0	1	3 states
0	1	0	4 states
0	1	1	5 states
1	0	0	6 states
1	0	1	7 states
1	1	0	8 states
1	1	1	9 states

Refresh cycle insertion intervals
See Table 3.7 (3)

Dummy cycle control

0	no dummy cycle
1	dummy cycle

Figure 3.7 (1) (a) Refresh Control Register

Don't use RMW instructions for DMEMCR1.

DMEMCR1 (009DH)

	7	6	5	4	3	2	1	0
bit Symbol	SRFC1	MACS1	BRM1	MACM1	MUXE1	MUXW11	MUXW10	MAC1
Read/Write	W	R / W						
After reset	1	0	0	0	0	0	0	0
Functions	Self-refresh 0 : self-refresh 1 : self-refresh release	Memory access 0: 2$\overline{CAS}$ 1: 2$\overline{WE}$	Bus release mode control 0 : release 1 : not release	Memory access control 0 : normal 1 : slow	Address multiplexing 0 : disable 1 : enable	Multiplex address length 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		Memory access control 0 : disable 1 : enable

Memory access control

0	disable
1	enable

Multiplex address length control
(valid only when MUXE1 = "1")

MUXW11	MUXW10	Multiplex address length
0	0	8 bit
0	1	9 bit
1	0	10 bit
1	1	11 bit

Address multiplex control

0	disable
1	enable

Memory access control

0	normal access mode
1	slow access mode

Bus release mode control

0	Also releases DRAM control signals.
1	Does not release DRAM control signals.

Memory access (*)

0	2$\overline{CAS}$
1	2$\overline{WE}$

Self-refresh control

0	self-refresh
1	self-refresh release

* <MACS1> is valid only with a 16-bit bus.

Figure 3.7 (1) (b) DRAM Memory Access Control Register

Control register

DREFCR3 (009EH)

	7	6	5	4	3	2	1	0
bit Symbol	DMI3	RS32	RS31	RS30	RW32	RW31	RW30	RC3
Read/Write	R / W							
After reset	0	0	0	0	0	0	0	0
Functions	Dummy cycle 0 : no dummy cycle 1 : dummy cycle	Refresh cycle insertion interval 000: 31 states 001: 62 states 010: 78 states 011: 97 states 100: 109 states 101: 124 states 110: 154 states 111: 195 states			Refresh cycle width 000: 2 states 001: 3 states 010: 4 states 011: 5 states 100: 6 states 101: 7 states 110: 8 states 111: 9 states			Refresh cycle 0 : no refresh cycle 1 : refresh cycle

Refresh cycle control

0	no refresh cycle
1	refresh cycle

Refresh cycle width control

RW32	RW31	RW30	Refresh cycle width
0	0	0	2 states
0	0	1	3 states
0	1	0	4 states
0	1	1	5 states
1	0	0	6 states
1	0	1	7 states
1	1	0	8 states
1	1	1	9 states

Refresh cycle insertion interval
See Table 3.7 (3).

Dummy cycle control

0	no dummy cycle
1	dummy cycle

Figure 3.7 (2) (a) Refresh Control Register

Don't use RMW instructions for DMEMCR1.

DMEMCR3 (009FH)

	7	6	5	4	3	2	1	0
bit Symbol	SRFC3	MACS3	BRM3	MACM3	MUXE3	MUXW31	MUXW30	MAC3
Read/Write	W	R / W						
After reset	1	0	0	0	0	0	0	0
Functions	Self-refresh 0 : self-refresh 1 : self-refresh release	Memory access 0 : 2$\overline{CAS}$ 1 : 2$\overline{WE}$	Bus release mode control 0 : release 1 : not release	Memory access control 0 : normal 1 : slow	Address multiplexing 0 : disable 1 : enable	Multiplex address length 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		Memory access control 0 : disable 1 : enable

Memory access control

0	disable
1	enable

Multiplex address length control
(valid only when MUXE3 = "1")

MUXW31	MUXW30	Multiplex address length
0	0	8 bit
0	1	9 bit
1	0	10 bit
1	1	11 bit

Address multiplex control

0	disable
1	enable

Memory access control

0	normal access mode
1	slow access mode

Bus release mode control

0	Also releases DRAM control signal.
1	Does not release DRAM control signal.

Memory access (*)

0	2$\overline{CAS}$
1	2$\overline{WE}$

Self-refresh control

0	self-refresh
1	self-refresh release

* <MACS3> is valid only with a 16-bit bus.

Figure 3.7 (2) (b) DRAM Memory Access Control Register

Description of Operation

TMP95C063 has a two-channel (DRAM1, DRAM3) internal DRAM controller. The two channels are normally linked to CS1 and CS3 of the CS wait controller. The DRAM controller generates the DRAM access cycle. The two channels are independent and can use different access modes. The DRAM signals share pins with port 6 (for details on setting the pins to DRAM pins, see 3.5.4, Port 6). The access mode automatically determines the function of the shared pins. Table 3.7 (1) shows the shared pins. As both channels operate identically, the following describes channel 1 (DRAM1) only.

(1) Memory Access Control

Setting DMEMCR1<MAC1> to "1" enables access control. If the area set as the $\overline{CS1}$ area in the CS wait controller is accessed when access control is enabled, a valid signal is output to DRAM in accordance with the DRAM memory access control register setting. The access cycle (bus cycle, number of waits) at this time depends on the $\overline{CS1}$ area setting in the CS wait controller.

If the bus size is 16 bits, the specified area is accessed using either the $\overline{2CAS}$ ($\overline{RAS}$, $\overline{UCAS}$, $\overline{LCAS}$, $\overline{WE}$) or the $\overline{2WE}$ mode ($\overline{RAS}$, $\overline{CAS}$, $\overline{UW}$, $\overline{LW}$), depending on the DMEMCR1<MACS1> setting. If <MACS1> is set to "0", the $\overline{2CAS}$ mode is used. If <MACS1> is set to "1", the $\overline{2WE}$ mode is used. A reset clears <MACS1> to "0", specifying the $\overline{2CAS}$ mode.
When the bus size is 8 bits, the specified area is accessed by the $\overline{RAS}$, $\overline{CAS}$, and $\overline{WE}$ signals regardless of the <MACS1> setting.

To facilitate the connection with low-speed DRAM, the DRAM controller accelerates the rising of the $\overline{RAS}$ signal when a wait is inserted, and extends the $\overline{RAS}$ pre-charge time (RAS high width). Slow access mode is set by DMEMCR1<MACM1>. A reset clears <MACH1> to "0" and sets normal mode.

The internal address multiplexer outputs the row/column address from A0-A11 during the access cycle. The DMEMCR <MUXE> bit specifies whether or not to multiplex addresses, and DMEMCR <MUXW0, 1> specifies the multiplexed address width. Note, however, that the multiplexed address lines depend on the bus size: 8-bit or 16-bit. Table 3.7 (2) shows the correspondence between the multiplexed address width and access bus size.

Figures 3.7 (3) and 3.7 (4) are the access timing charts.

Table 3.7 (1) DRAM Pins

Pin Name \ Mode	8-Bit Bus	16-Bit Bus	
		2$\overline{CAS}$ Mode	2$\overline{WE}$ Mode
P60 ($\overline{CS1}$/$\overline{RAS1}$)	$\overline{RAS1}$	$\overline{RAS1}$	$\overline{RAS1}$
P61 ($\overline{CAS1}$/$\overline{WE1}$)	$\overline{CAS1}$	$\overline{WE1}$	$\overline{CAS1}$
P62 ($\overline{LCAS1}$/$\overline{LW1}$/$\overline{REFOUT1}$)	$\overline{REFOUT1}$	$\overline{LCAS1}$	$\overline{LW1}$
P63 ($\overline{UCAS1}$/$\overline{UW1}$/$\overline{WE1}$)	$\overline{WE1}$	$\overline{UCAS1}$	$\overline{UW1}$
P64 ($\overline{CS3}$/$\overline{RAS3}$)	$\overline{RAS3}$	$\overline{RAS3}$	$\overline{RAS3}$
P65 ($\overline{CAS3}$/$\overline{WE3}$)	$\overline{CAS3}$	$\overline{WE3}$	$\overline{CAS3}$
P66 ($\overline{LCAS3}$/$\overline{LW3}$/$\overline{REFOUT3}$)	$\overline{REFOUT3}$	$\overline{LCAS3}$	$\overline{LW3}$
P67 ($\overline{UCAS3}$/$\overline{UW3}$/$\overline{WE3}$)	$\overline{WE3}$	$\overline{UCAS3}$	$\overline{UW3}$

Table 3.7 (2) Address Multiplexing

Row Address	Column Address							
	8 BIT		9 BIT		10 BIT		11 BIT	
	8	16	8	16	8	16	8	16
A0	A8	–	A9	–	A10	–	A11	–
A1	A9	A9	A10	A10	A11	A11	A12	A12
A2	A10	A10	A11	A11	A12	A12	A13	A13
A3	A11	A11	A12	A12	A13	A13	A14	A14
A4	A12	A12	A13	A13	A14	A14	A15	A15
A5	A13	A13	A14	A14	A15	A15	A16	A16
A6	A14	A14	A15	A15	A16	A16	A17	A17
A7	A15	A15	A16	A16	A17	A17	A18	A18
A8	–	A16	A17	A17	A18	A18	A19	A19
A9	–	–	–	A18	A19	A19	A20	A20
A10	–	–	–	–	–	A20	A21	A21
A11	–	–	–	–	–	–	–	A22

multiplex address length

access bus size (set in the CS wait controller)

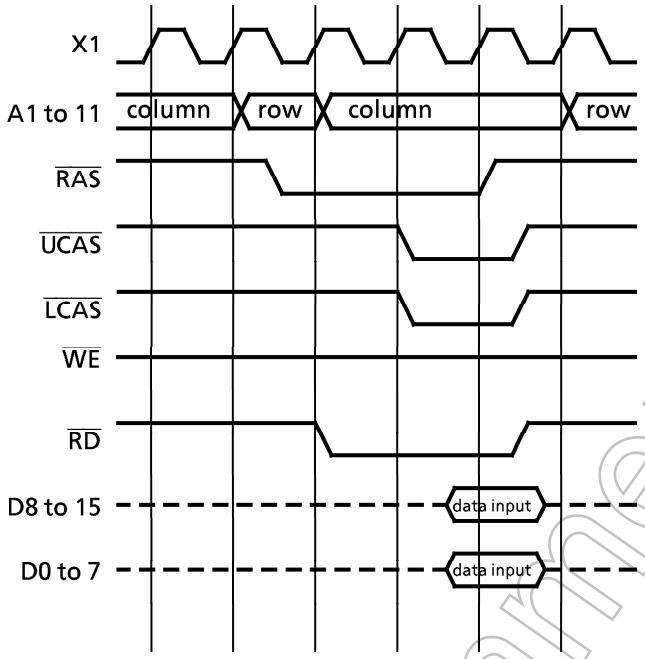


Figure 3.7 (3) (a) DRAM Access Timing (2$\overline{CAS}$ Mode and Read Cycle)

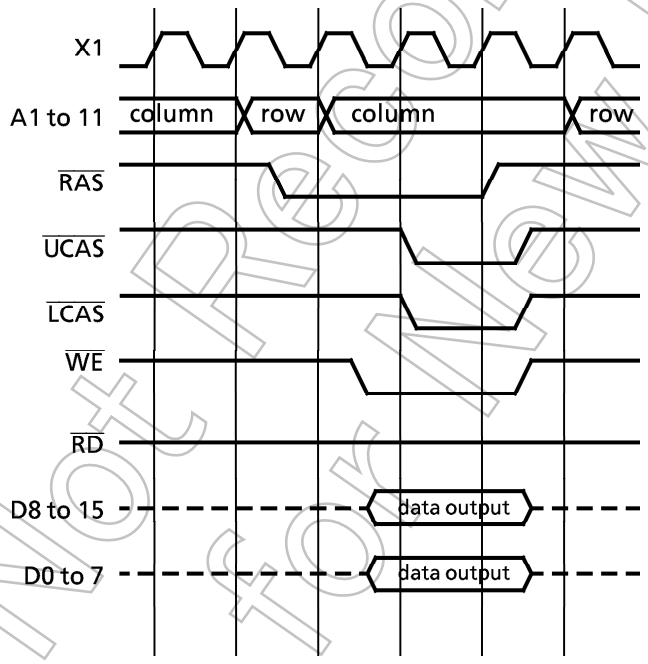


Figure 3.7 (3) (b) DRAM Access Timing (2$\overline{CAS}$ Mode and Word Access)

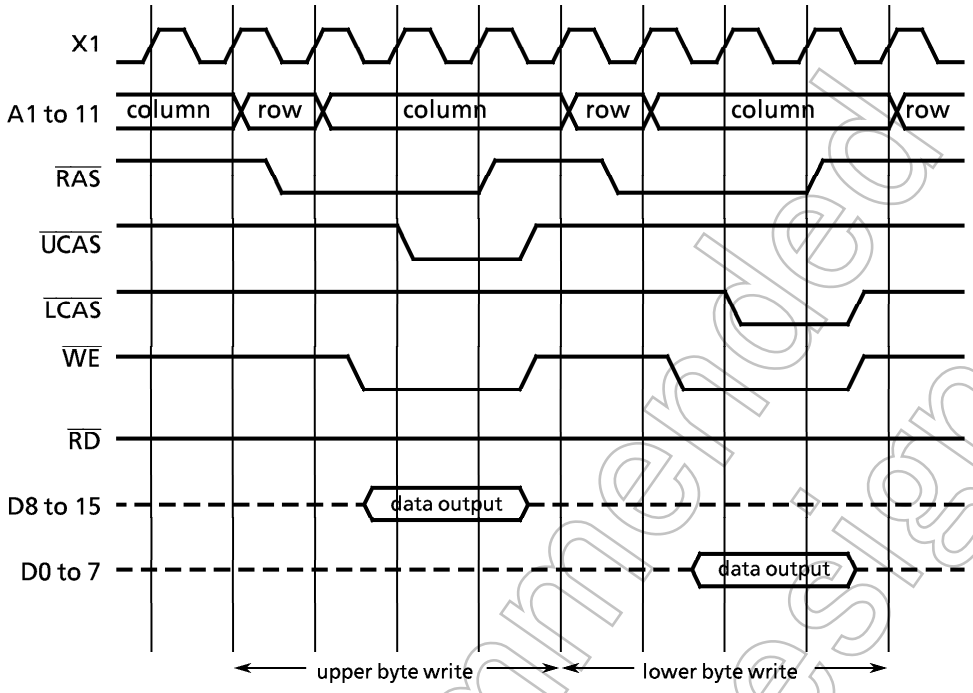


Figure 3.7 (3) (c) DRAM Access Timing (2$\overline{\text{CAS}}$ Mode, Write Cycle, and Byte Access)

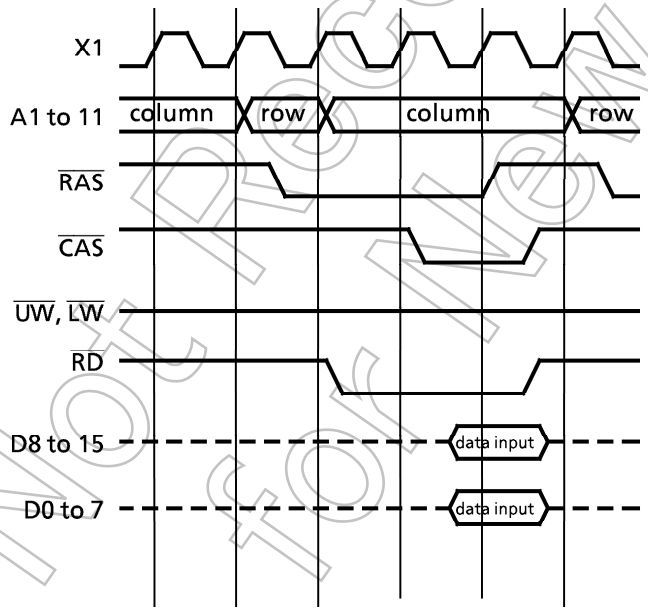


Figure 3.7 (4) (a) DRAM Access Timing (2$\overline{\text{WE}}$ Mode and Read Cycle)

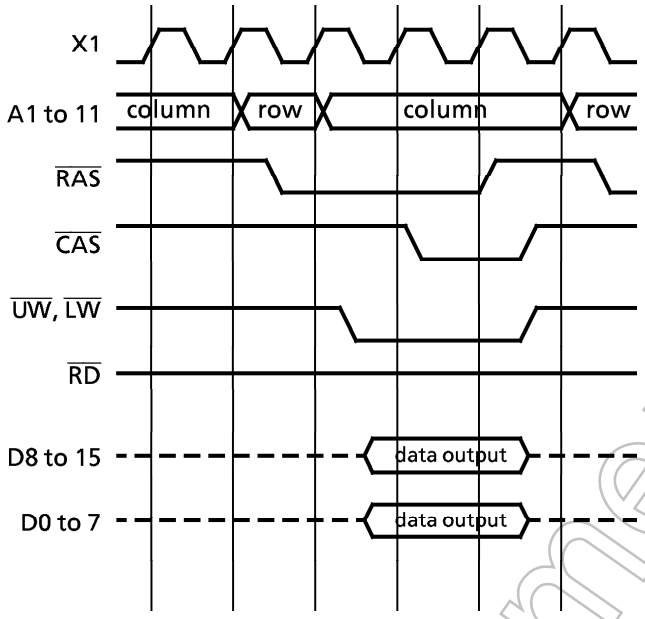


Figure 3.7 (4) (b) DRAM Access Timing (2WE Mode, Write Cycle, and Word Access)

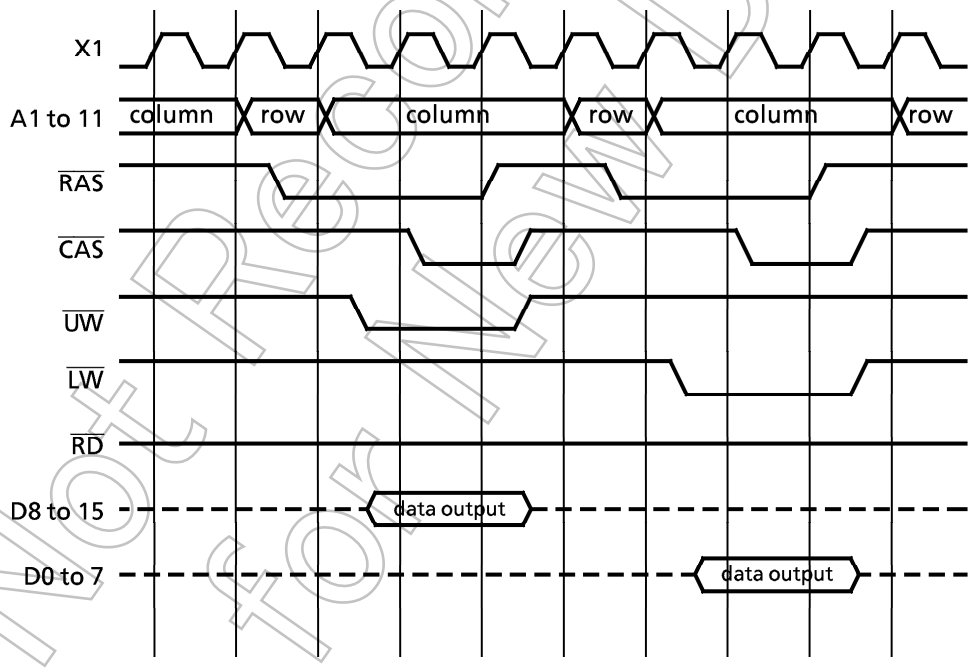


Figure 3.7 (4) c DRAM Access Timing (2WE Mode, Write Cycle, and Byte Access)

(2) Refresh Controller Block

TMP95C063 outputs the $\overline{RAS}$ / $\overline{CAS}$ ($\overline{LCAS}$ / $\overline{UCAS}$) signals, which can be used for refreshing DRAM. When using an 8-bit bus, the device also outputs state signal $\overline{REFOUT}$ to indicate a refresh cycle.

As the output cycle and pulse width of the $\overline{RAS}$ / $\overline{CAS}$ ($\overline{LCAS}$ / $\overline{UCAS}$) output can be set by program, the DRAM refresh is easily realized.

The refresh controller block has the following features.

- Refresh modes : $\overline{CAS}$-before-$\overline{RAS}$ interval refresh mode,
  $\overline{CAS}$-before-$\overline{RAS}$ self-refresh mode
- Refresh interval : 31-195 states (programmable)
- Refresh cycle width : 2-9 states (programmable)
- Dummy cycles can be generated.
- The refresh cycle is asynchronous to the CPU operating cycle.

i) $\overline{CAS}$-before-$\overline{RAS}$ interval refresh mode

The refresh interval and the refresh cycle width in the $\overline{CAS}$-before-$\overline{RAS}$ interval refresh mode vary according to the DRAM being used.

The refresh interval and the refresh cycle width in TMP95C063 can be set in accordance with the system clock and type of DRAM used, by modifying the value of the refresh control register.

Figure 3.7 (5) shows examples of $\overline{CAS}$-before-$\overline{RAS}$ refresh cycle timings.

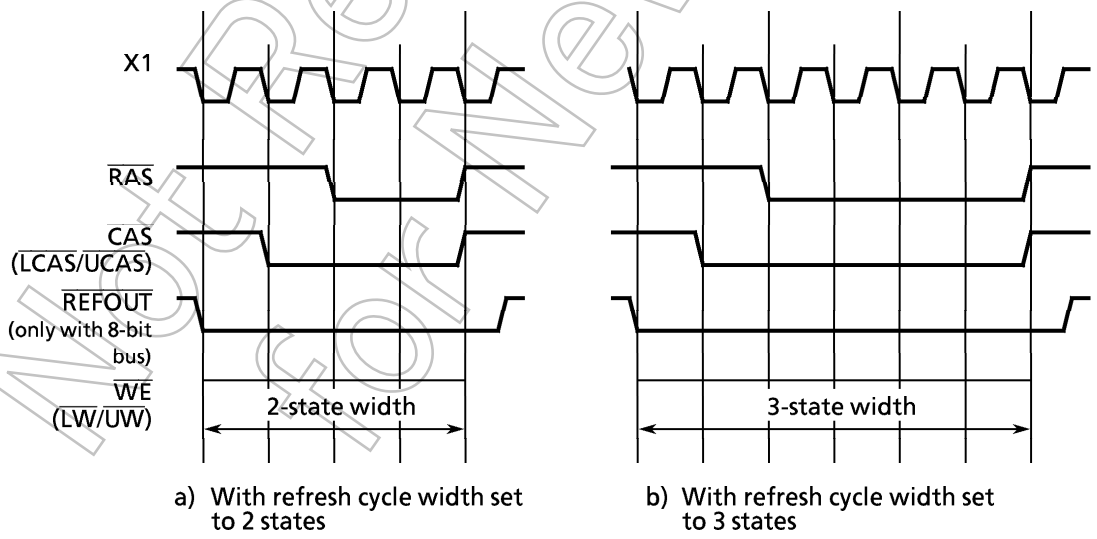


Figure 3.7 (5) Refresh Cycle Timing Examples

The following describes how to set the registers.

Figure 3.7 (1) (a) shows the bit configuration of refresh control register DREFCR1.

① Refresh cycle insertion interval

Bit 3 of the DREFCR1<RS12 to 10> register is used to set the insertion interval in accordance with the system clock used.

Example: When using the system clock at 25 MHz, set this bit to "111" to set the DRAM refresh cycle to 15.6 μs.

Table 3.7 (3) Refresh Cycle Insertion Interval

Refresh Cycle			Insertion Interval (States)	Frequency (f_{OSC})						
RS12	RS11	RS10		8 MHz	10 MHz	12.5 MHz	14 MHz	16 MHz	20 MHz	25 MHz
0	0	0	31	7.55	6.2	4.96	4.43	3.88	3.1	2.5
0	0	1	62	15.5	12.4	9.92	8.86	7.75	6.2	5.0
0	1	0	78	19.5	15.6	12.48	11.14	9.75	7.8	6.2
0	1	1	97	24.25	19.4	15.52	13.86	12.13	9.7	7.7
1	0	0	109	27.25	21.8	17.44	15.57	13.63	10.9	8.7
1	0	1	124	31.0	24.8	19.84	17.72	15.5	12.4	9.9
1	1	0	154	38.5	30.8	24.7	22.0	19.3	15.4	12.3
1	1	1	195	48.75	39.0	31.2	27.86	24.4	19.5	15.6

(Unit : μs)

② Refresh Cycle Width

Bit 3 of the DREFCR<RW12 to 10> register can vary the refresh cycle width ($\overline{RAS}$, $\overline{CAS}$ low output width). (2-9 states)

③ Refresh cycle control

Manipulating the bits of the DREFCR<RC1> register enables or disables the refresh cycle.

ii) $\overline{CAS}$-before-$\overline{RAS}$ self-refresh mode

This mode is used when the clock supplied to the DRAM controller is stopped by a HALT instruction (IDLE, STOP) while refreshing using the $\overline{CAS}$-before-$\overline{RAS}$ interval refresh mode.

Figure 3.7 (6) is the timing chart for self-refresh mode.

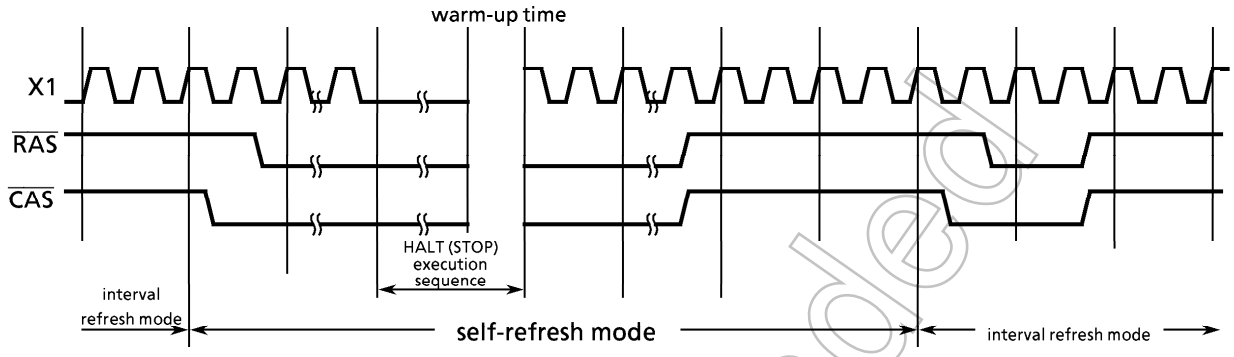


Figure 3.7 (6) Self-Refresh Cycle Timing

To refresh DRAM in $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ self-refresh mode, first, set DRAM to $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh mode. Then, before entering the HALT instruction, set DMEMCR1<SRFC1> to "0" to execute a single $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh. Then, the $\overline{\text{CAS}}$ and $\overline{\text{RAS}}$ pins maintain their low levels, and $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ self-refresh mode starts. When the HALT is released and the clock is supplied to the DRAM controller, DMEMCR1<SRFC1> is automatically set to "1" and $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ self-refresh mode is released. After the release, be sure to execute a single $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh to return to interval refresh mode. (Note that when a HALT is released by a reset, the I/O registers are initialized; therefore, the $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh is not executed.)

After setting DMEMCR1<SRFC1> to "0", execute any instruction, such as a NOP instruction, then execute a HALT instruction.

(3) DRAM Initialization

The DRAM controller can generate the continuous $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ dummy cycles required when using DRAM. Setting the DREFCR1<DMI1> bit to "1" generates the dummy cycles. Dummy cycle generation is released by writing "0" to <DMI1> (including a write due to a reset), by enabling refresh cycle insertion (DREFCR1<RC1>="1"), or by enabling access control (DMEMCR1<MAC1>="1").

When dummy cycle generation is released by enabling refresh cycle insertion or by enabling access control, the <DMI1> bit is not cleared to zero. The dummy cycle width is fixed to 4 states; the interval, to 6 states.

Figure 3.7 (7) is the timing chart for the generation of $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ dummy cycles.

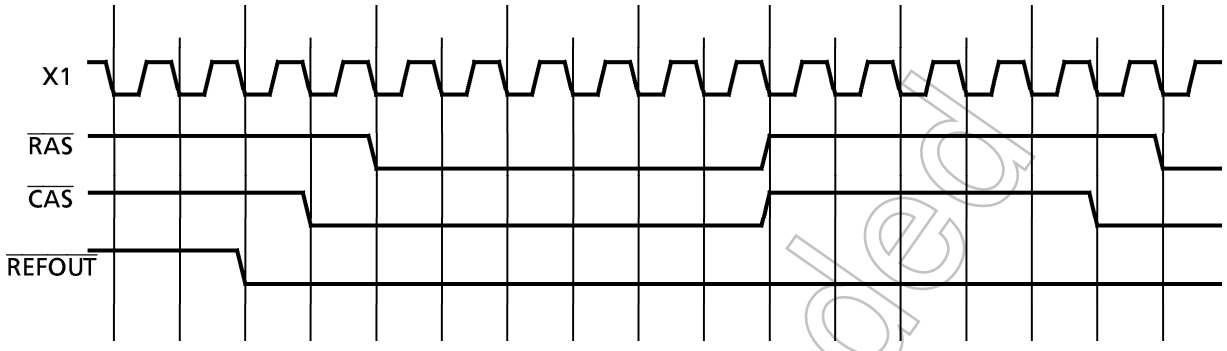


Figure 3.7 (7) CAS-Before-RAS Dummy Cycle Timing

(4) Priorities

As the DRAM refresh cycle is asynchronous to the CPU operating cycle, the refresh cycle may overlap with DRAM read and write cycles. If an overlap occurs, the DRAM controller gives priority to the cycle that started first. If the refresh cycle is given priority, the DRAM controller automatically inserts wait states in the memory access cycle until the refresh cycle completes.

(5) Bus Release Mode

TMP95C063 has a bus release function. Select the mode for the DRAM control pins using DMEMCR1<BRM1>. Set either to release mode (set to high impedance) in the same way as other pins, or to non-release mode (remain active when the bus is released). The latter mode only supports output of refresh cycles. For details on the states of the other pins when the bus is released, see 3.15 (2), Pin States at Bus Release.

(i) DRAM Control Pin Release Mode (DMEMCR1<BRM1> = 0)

When the bus release request pin (BUSRQ) is set to active (low), TMP95C063 acknowledges a bus release request. When the current bus cycle (including a DRAM access cycle) completes, TMP95C063 first sets the DRAM control pins to high, then turns the output buffer off to set the pins to high impedance. As the refresh cycle is asynchronous to the access cycle, when a refresh request is generated and has to wait because of contention with an access cycle preceding the timing of a bus release request, the refresh cycle is generated and the bus release timing is delayed until the refresh cycle completes.

The refresh counter continues to count when the bus is released. Only one refresh request generated during the bus release is held. The refresh cycle is generated immediately upon return of the bus mastership to TMP95C063 at bus release completion.

As the bus release request and the refresh counter are asynchronous to the bus cycle, in this mode, refresh cycles must be generated by the external bus master during bus release.

(ii) DRAM Control Pin Non-Release Mode (DMEMCR1<BRM1> = "1")

This mode is valid when DRAM is not accessed by the external bus master during bus release. When this mode is set, the DRAM pins do not release the bus when a bus release request occurs. The pins continue to operate but support refresh cycles only. However, all other pins are released. Unlike mode (i), the bus release timing is not affected by refresh requests.

A reset clears DMECR1<BRM1> to "0" selecting DRAM control pin release mode.

(6) Notes

When refresh and access contend, the $\overline{WR}$ and $\overline{HWR}$ pins are set to active and output refresh signals. (Figure 3.7 (8))

TMP95C063F does not support DRAM set to write-per-bit mode at timing (a) in Figure 3.7 (8).

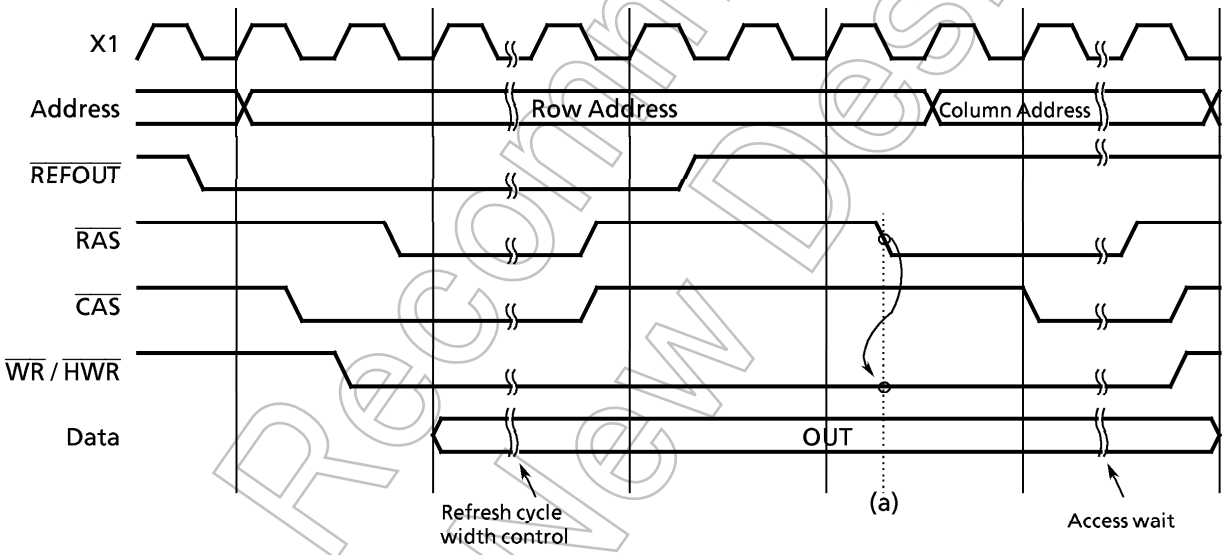
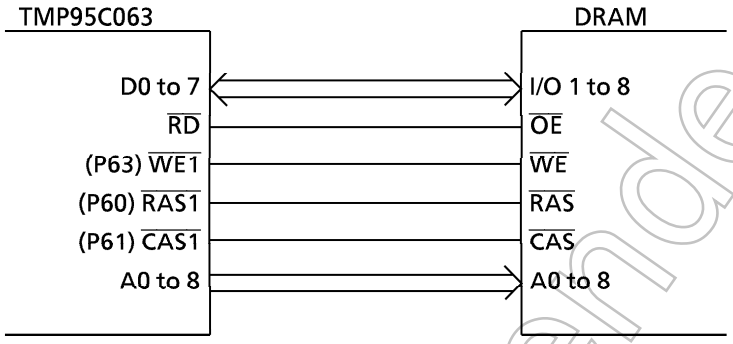


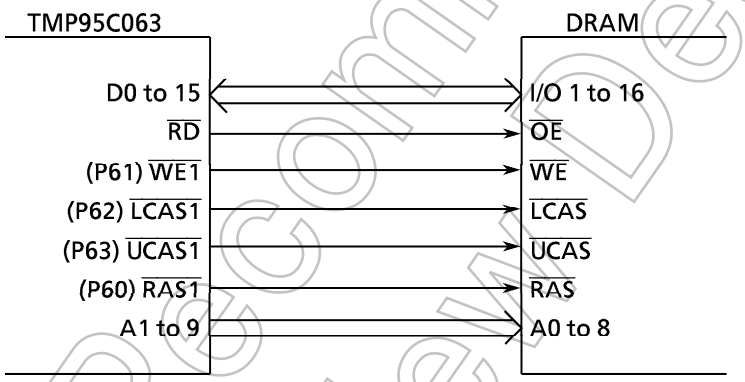
Figure 3.7 (8) Timings for Refresh and Access Contention

(7) Connection Example

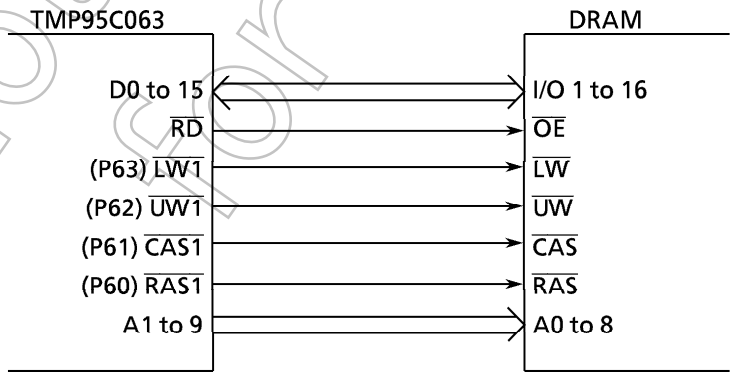
Connection Example (1) 8-Bit Bus Configuration



Connection Example (2) 16-Bit Bus Configuration ($\overline{2CAS}$ Mode)



Connection Example (3) 16-Bit Bus Configuration ($\overline{2WE}$ Mode)



3.8 8-Bit Timers

TMP95C063 incorporates eight 8-bit timers (timers 0 to 7). Each timer can operate independently or be cascaded to form four 16-bit timers. The 8-bit timers have the following four operating modes.

- 8-bit interval timer mode (8 channels)
- 16-bit interval timer mode (4 channels)

} The above two modes can be combined (for example, four 8-bit timers and two 16-bit timers)

- 8-bit programmable square wave (PPG : variable cycle, variable duty) output mode (4 channels)
- 8-bit PWM (pulse width modulation: variable duty at fixed cycle) output mode (4 channels)

Figure 3.8 (1) is a block diagram for 8-bit timers (timers 0, 1).

Timers 2 and 3, timers 4 and 5, and timers 6 and 7 have the same circuit configuration as timers 0 and 1.

Each interval timer consists of an 8-bit up-counter, an 8-bit comparator, and an 8-bit timer register. One timer flip-flop each (TFF1, TFF3, TFF5, and TFF7) is provided for the timer pairs: timers 0 and 1, timers 2 and 3, timers 4 and 5, and timers 6 and 7.

Of the input clock sources for interval timers, the ϕT1, ϕT4, ϕT16, and ϕT256 internal clocks are obtained from the 9-bit prescaler shown in Figure 3.8 (2).

The 8-bit timer operating mode and the timer flip-flops are controlled by nine control registers (T01MOD, T23MOD, T45MOD, T67MOD, T02FFCR, T46FFCR, T8RUN, T16RUN, and TRDC)

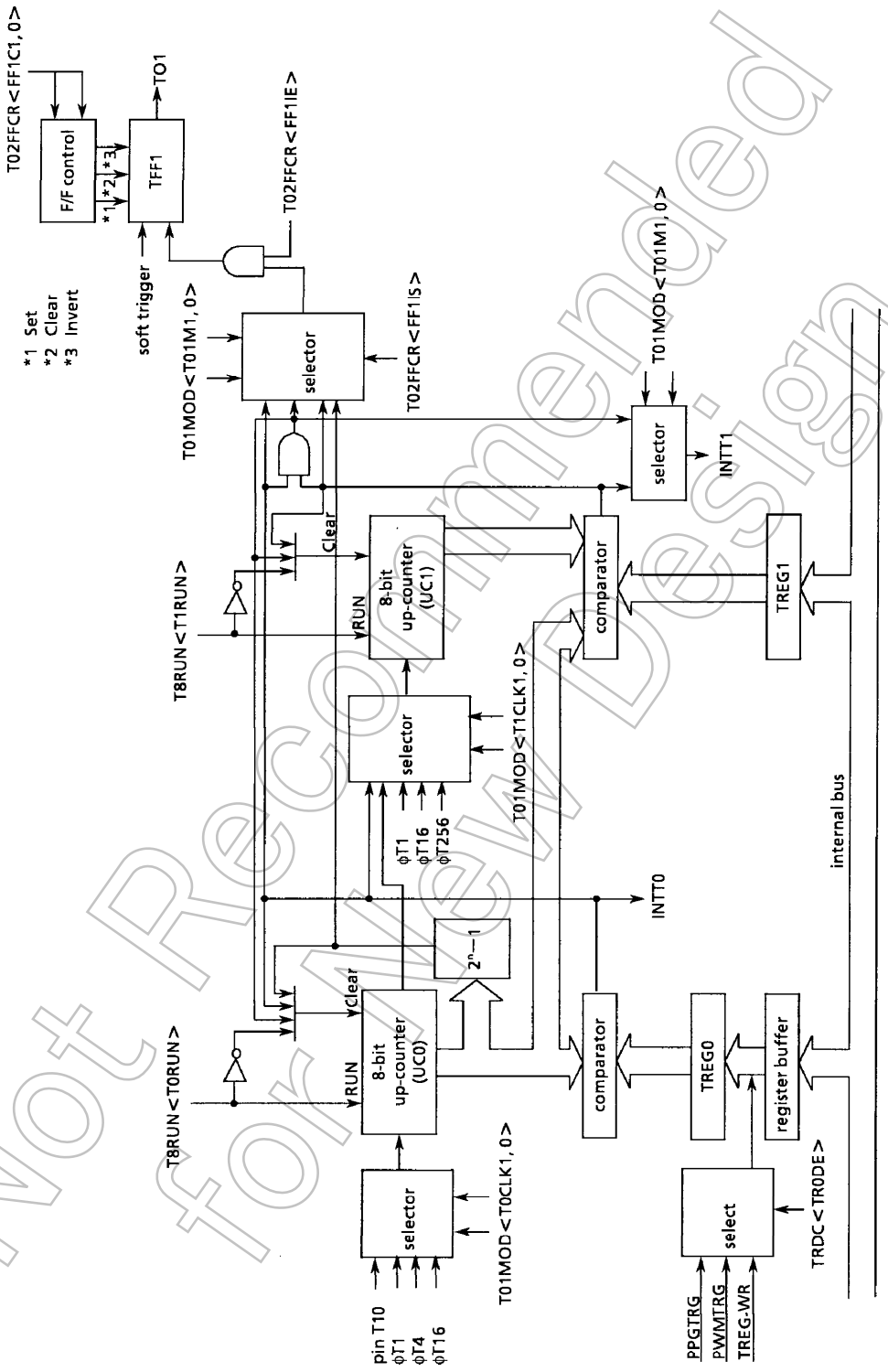


Figure 3.8 (1) 8-Bit Timer Block Diagram (Timers 0,1)

① Prescaler

The input to the 9-bit prescaler is the CPU fundamental clock (fc) divided by four (fc/4). The prescaler generates an input clock for the 8-bit timers, the 16-bit timer/event counters, and baud rate generator, for example.

The 8-bit timers can use the following four clock signals: ϕT1, ϕT4, ϕT16, and ϕT256.

To set the prescaler to count or stop, use timer control register T16RUN<PRRUN>. Setting T16RUN<PRRUN> to "1" starts the count. Clearing <PRRUN> to "0" clears and stops the prescaler. Resetting clears <PRRUN> to "0", and clears and stops the prescaler.

Input clock \ fc	Interval 25 MHz
ϕT1 (8/fc)	320 ns
ϕT4 (32/fc)	1.28 μs
ϕT16 (128/fc)	5.12 μs
ϕT256 (2048/fc)	81.92 μs

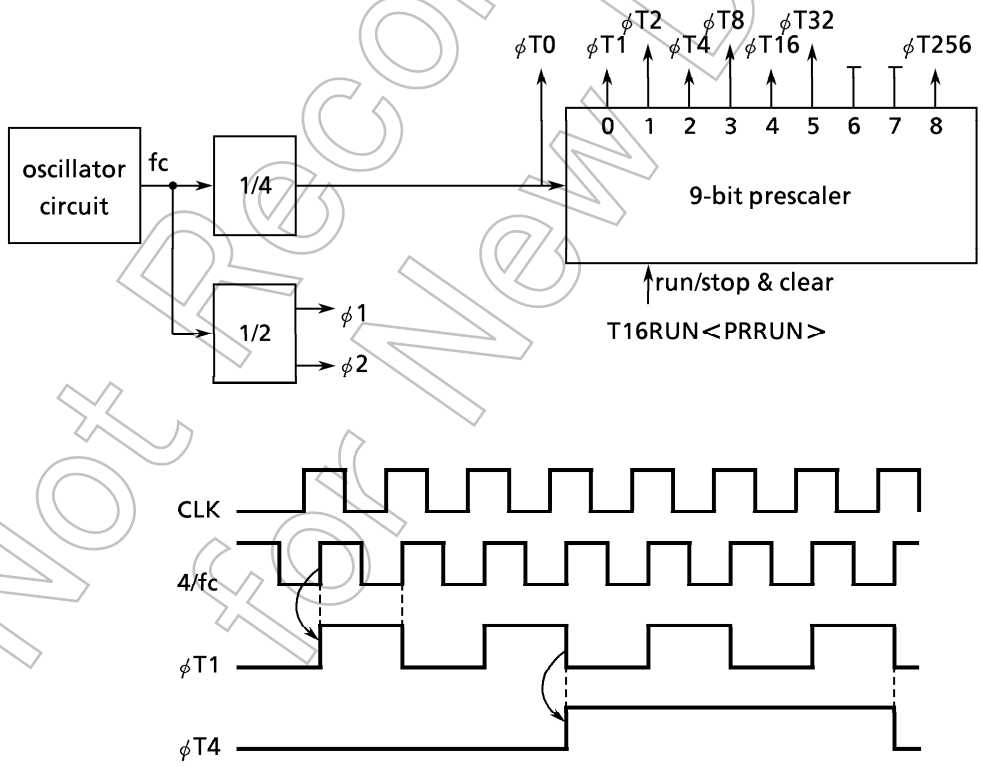


Figure 3.8 (2) Prescaler

② Up-counter

The up-counter is an 8-bit binary counter that counts up using the input clock specified by timer 0 and 1 mode registers T01MOD, T23MOD, T45MOD, and T67MOD.

The timer 0, 2, 4, and 6 input clocks are selected from internal clocks ϕT1, ϕT4, and ϕT16, and from external clocks output from pins T10, T12, T14, and T16 in accordance with the T01MOD, T23MOD, T47MOD, and T67MOD settings.

The timer 1, 3, 5, and 7 input clocks vary according to the operating mode. When the up-counter is set to 16-bit timer mode, timer 0, 2, 4, and 6 overflow output is used as an input clock.

When the up-counter is set to other than 16-bit timer mode, two further settings are available: internal clocks ϕT1, ϕT16, or ϕT256 based on the T01MOD, T23MOD, T45MOD, and T67MOD settings, and timer 0, 2, 4 and 6 comparator output (match detect).

Example : If T01MOD<T01M1,0> is set to "01", the timer 0 overflow output is used as the timer 1 input clock (16-bit timer mode).
If T01MOD7,6 is "00" and T01MOD3,2 is "01", ϕT1 is used as the timer 1 input clock (8-bit timer mode).

The T01MOD, T23MOD, T45MOD, and T67MOD registers also set the operating mode. A reset sets the up-counter to 8-bit timer mode.

To control the count, stop, and clear functions of each up-counter interval timer, use timer control register T8RUN. A reset clears all up-counters and stops the timers.

③ Timer registers

The timer registers are 8-bit registers for setting interval times. When the setting of timer registers TREG0-7 matches the up-counter value, the comparator match detect signal becomes active. If "00H" is set, the match detect signal is activated when the up-counter overflows.

Timer registers TREG0, 2, 4, and 6 have a double-buffer configuration and are paired with a register buffer.

TREG0, 2, 4 and 6 enable or disable the double-buffer using timer register double-buffer control register TRDC<TR0/2/4/6DE>. Setting <TR0/2/4/6DE> to "0" disables the double-buffer; setting <TR0/2/4/6DE> to "1" enables the double-buffer.

With the double-buffer enabled, data are transferred from the register buffer to the timer register at a 2^n-1 overflow in pulse width modulation (PWM) mode, or at an interval comparison match in programmable pulse generation (PPG) mode.

A reset initializes <TR0/2/4/6DE> to "0", disabling the double-buffer. When using the double-buffer, first write data to the timer register and set <TR0/2/4/6DE> to "1", then write the following data to the register buffer.

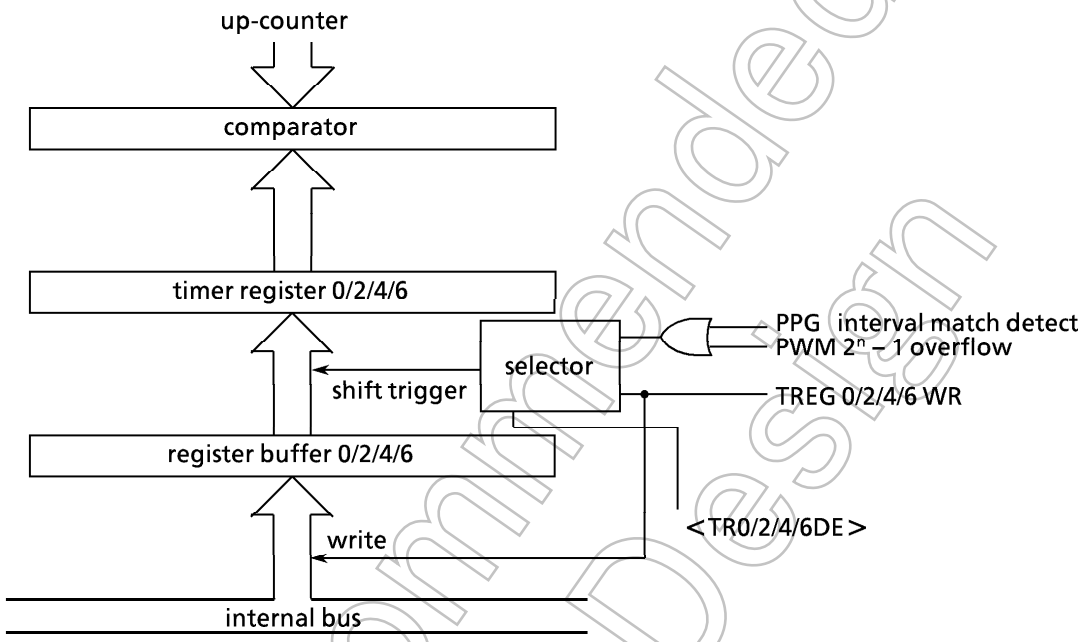


Figure 3.8 (3) Timer Register 0/2/4/6 Configuration

Note : The timer register and register buffer are allocated to the same address in memory. When <TR0/2/4/6DE> is set to "0", the same value is written to both the register buffer and the timer register. When <TR0/2/4/6DE> is set to "1", the value is written to the register buffer only.

The timer registers are allocated in memory as follows.

TREG0 :	000022H	TREG4 :	000029H
TREG1 :	000023H	TREG5 :	00002AH
TREG2 :	000026H	TREG6 :	00002DH
TREG3 :	000027H	TREG7 :	00002EH

All registers are write-only; cannot read data from them.

As the initial values are undefined, when using an 8-bit timer, be sure to write values.

T01MOD (0024H)

	7	6	5	4	3	2	1	0
bit Symbol	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: $2^6 - 1$ PWM 10: $2^7 - 1$ interval 11: $2^8 - 1$		00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI0 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 0 input clock

00	External clock TI0
01	Internal clock ϕT1
10	Internal clock ϕT4
11	Internal clock ϕT16

Timer 1 input clock

	T01MOD7, 6 ≠ 01	T01MOD7, 6 = 01
00	Timer 0 comparator output	Timer 0 overflow output
01	Internal clock ϕT1	
10	Internal clock ϕT16	(16-bit timer mode)
11	Internal clock ϕT256	

PWM0 interval selection
(Except for PWM mode, don't care)

00	——
01	$2^6 - 1$
10	$2^7 - 1$
11	$2^8 - 1$

Timer 0 and 1 operating mode selection

00	8-bit timer x 2 (timers 0,1)
01	16-bit timer
10	8-bit programmable square wave output
11	8-bit PWM output (timer 0) + 8-bit timer (timer 1)

Figure 3.8 (4) Timer 0/1 Mode Register (T01MOD)

T23MOD (0028H)

	7	6	5	4	3	2	1	0
bit Symbol	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: $2^6 - 1$ PWM 10: $2^7 - 1$ interval 11: $2^8 - 1$		00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI2 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 2 input clock

00	External clock TI2
01	Internal clock ϕT1
10	Internal clock ϕT4
11	Internal clock ϕT16

Timer 3 input clock

	T23MOD7, 6 ≠ 01	T23MOD7, 6 = 01
00	Timer 2 comparator output	Timer 2 overflow output
01	Internal clock ϕT1	
10	Internal clock ϕT16	(16-bit timer mode)
11	Internal clock ϕT256	

PWM2 interval selection
(Except for PWM mode, don't care)

00	——
01	$2^6 - 1$
10	$2^7 - 1$
11	$2^8 - 1$

Timer 2 and 3 operating mode selection

00	8-bit timer x 2 (timers 2, 3)
01	16-bit timer
10	8-bit programmable square wave output
11	8-bit PWM output (timer 2) + 8-bit timer (timer 3)

Figure 3.8 (5) Timer 0/1 Mode Register (T23MOD)

T45MOD (002BH)

	7	6	5	4	3	2	1	0
bit Symbol	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: 2^6 – 1 PWM 10: 2^7 – 1 interval 11: 2^8 – 1		00: TO4TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI4 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 4 input clock

00	External clock TI4
01	Internal clock ϕT1
10	Internal clock ϕT4
11	Internal clock ϕT16

Timer 5 input clock

	T45MOD7, 6 ≠ 01	T45MOD7, 6 = 01
00	Timer 4 comparator output	Timer 4 overflow output (16-bit timer mode)
01	Internal clock ϕT1	
10	Internal clock ϕT16	
11	Internal clock ϕT256	

PWM4 interval selection
(Except for PWM mode, don't care)

00	——
01	2^6 – 1
10	2^7 – 1
11	2^8 – 1

Timer 4 and 5 operating mode selection

00	8-bit timer x 2 (timers 4, 5)
01	16-bit timer
10	8-bit programmable square wave output
11	8-bit PWM output (timer 4) + 8-bit timer (timer 5)

Figure 3.8 (6) Timer 4/5 Mode Register (T45MOD)

T67MOD (002FH)

	7	6	5	4	3	2	1	0
bit Symbol	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: 2^6 – 1 PWM 10: 2^7 – 1 interval 11: 2^8 – 1		00: TO6TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI6 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 6 input clock

00	External clock TI6
01	Internal clock ϕT1
10	Internal clock ϕT4
11	Internal clock ϕT16

Timer 7 input clock

	T67MOD7, 6 ≠ 01	T67MOD7, 6 = 01
00	Timer 6 comparator output	Timer 6 overflow output
01	Internal clock ϕT1	
10	Internal clock ϕT16	(16-bit timer mode)
11	Internal clock ϕT256	

PWM6 interval selection
(Except for PWM mode, don't care)

00	——
01	2^6 – 1
10	2^7 – 1
11	2^8 – 1

Timer 6 and 7 operating mode selection

00	8-bit timer x 2 (timers 6, 7)
01	16-bit timer
10	8-bit programmable square wave output
11	8-bit PWM output (timer 6) + 8-bit timer (timer 7)

Figure 3.8 (7) Timer 6/7 Mode Register (T67MOD)

T02FFCR (0025H)

	TFF3				TFF1			
	7	6	5	4	3	2	1	0
bit Symbol	FF3C1	FF3C0	FF3IE	FF3IS	FF1C1	FF1C0	FF1IE	FF1IS
Read/Write	W		R/W		W		R/W	
After reset	–		0	0	–		0	0
Function	00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		1: TFF3 Invert Enable	0: Inverts by 8-bit timer 2. 1: Inverts by timer 3.	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		1: TFF1 Invert Enable	0: Inverts by 8-bit timer 0. 1: Inverts by timer 1.

Timer flip-flop TFF1 inversion signal selection
(Except for 8-bit timer mode, don't care)

0	Inverts by timer 0.
1	Inverts by timer 1.

Timer flip-flop TFF1 inversion

0	Disable
1	Enable

Timer flip-flop TFF1 control

00	Inverts TFF1 value (software inversion).
01	Sets TFF1 to "1".
10	Clears TFF1 to "0".
11	Don't care (Always read as "11")

Timer flip-flop TFF3 inversion signal selection
(Except for 8-bit timer mode, don't care)

0	Inverts by timer 2.
1	Inverts by timer 3.

Timer flip-flop TFF3 inversion

0	Disable
1	Enable

Timer flip-flop TFF3 control

00	Inverts TFF3 value (software inversion).
01	Sets TFF3 to "1".
10	Clear TFF3 to "0".
11	Don't care (Always read as "11")

Figure 3.8 (8) 8-Bit Timer Flip-Flop Control Register (T02FFCR)

T46FFCR (002CH)

	TFF7				TFF5			
	7	6	5	4	3	2	1	0
bit Symbol	FF7C1	FF7C0	FF7IE	FF7IS	FF5C1	FF5C0	FF5IE	FF5IS
Read/Write	W		R/W		W		R/W	
After reset	–		0	0	–		0	0
Function	00: Invert TFF7 01: Set TFF7 10: Clear TFF7 11: Don't care		1: TFF7 Invert Enable	0: Inverts by 8-bit timer 6. 1: Inverts by timer 7.	00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: Don't care		1: TFF5 Invert Enable	0: Inverts by 8-bit timer 4. 1: Inverts by timer 5.

Timer flip-flop TFF5 inversion signal selection (Except for 8-bit timer mode, don't care)

0	Inverts by timer 4.
1	Inverts by timer 5.

Timer flip-flop TFF5 inversion

0	Disable
1	Enable

Timer flip-flop TFF5 control

00	Inverts TFF5 value (software inversion).
01	Sets TFF5 to "1".
10	Clears TFF5 to "0".
11	Don't care (Always read as "11")

Timer flip-flop TFF7 inversion signal selection (Except for 8-bit timer mode, don't care)

0	Inverts by timer 6.
1	Inverts by timer 7.

Timer flip-flop TFF7 inversion

0	Disable
1	Enable

Timer flip-flop TFF7 control

00	Inverts TFF7 value (software inversion).
01	Sets TFF7 to "1".
10	Clear TFF7 to "0".
11	Don't care (Always read as "11")

Figure 3.8 (9) 8-Bit Timer Flip-Flop Control Register (T46FFCR)

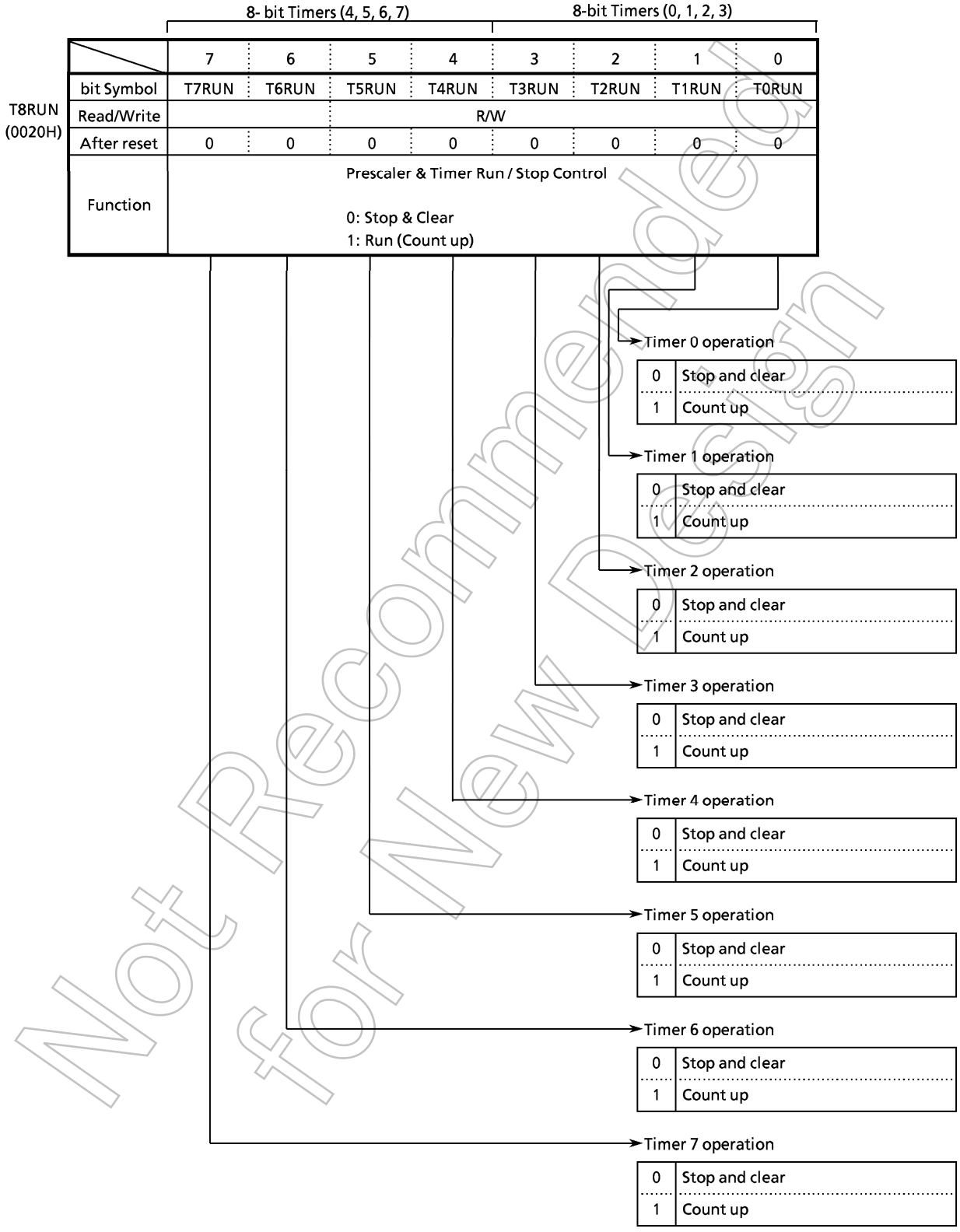


Figure 3.8 (10) 8-Bit Timer Operation Control Register (T8RUN)

T16RUN (003BH)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T9RUN	T8RUN				
Read/Write	R/W		R/W	R/W				
After reset	0		0	0				
Function	Prescaler & Timer Run/Stop CONTROL 0: Stop & Clear 1: Run (Count up)							

16-bit timer (timer 8) operation

0	Stop and clear
1	Count up

16-bit timer (timer 9) operation

0	Stop and clear
1	Count up

Prescaler operation

0	Stop and clear
1	Count up

Figure 3.8 (11) 16-Bit Timer Operation Control Register (T16RUN)

TRDC (0021H)

	7	6	5	4	3	2	1	0
bit Symbol					TR6DE	TR4DE	TR2DE	TR0DE
Read/Write					R/W			
After reset					0	0	0	0
Function					0: Duble Buffer Disable 1: Duble Buffer Enable			

Timer register 0 double-buffer control

0	Disable
1	Enable

Timer register 2 double-buffer control

0	Disable
1	Enable

Timer register 4 double-buffer control

0	Disable
1	Enable

Timer register 6 double-buffer control

0	Disable
1	Enable

Figure 3.8 (12) Timer Register Double-Buffer Control Register (TRDC)

④ Comparator

The comparator compares the up-counter value with the timer register value. If the values match, the comparator clears the up-counter to 0 and generates an interrupt (INTT0 to 7). If the timer flip-flop invert is enabled at this time, the comparator inverts the timer flip-flop value.

⑤ Timer flip-flops (timer F/F)

Each interval timer match detect signal (comparator output) inverts the timer flip-flops and outputs the values to timer output pins TO1 (also used as P91), TO3 (also used as P93), TO5 (also used as P95), and TO7 (also used as P97).

One timer flip-flop is provided for a timer pair: TFF1 for timer pair 0,1; TFF3 for pair 2,3; TFF5 for pair 4,5; and TFF7 for pair 6,7. TFF1 is output to pin TO1, TFF3 to pin TO3, TFF5 to pin TO5, and TFF7 to pin TO7.

The following explains the operation of the 8-bit timers.

(1) 8-bit Timer Mode

Eight interval timers 0 to 7 can be used independently as 8-bit interval timers. As all the timers operate the same, the following describes timer 1 only.

① Generating a fixed-interval interrupt

When using timer 1 to generate a timer 1 interrupt (INTT1) for each fixed interval, first halt timer 1, then set the operating mode, input clock, and interval in T01MOD and TREG1. Next, enable INTT1, and start timer 1 counting.

Example : If a timer 1 interrupt is required every 32 μs at fc = 25 MHz, set the registers in the following order:

```
                MSB          LSB
                 7 6 5 4 3 2 1 0
T8RUN    ←      - - - - - - 0 -     Stops timer 1 and clears it to "0".
T01MOD   ←      0 0 X X 0 1 - -     Sets 8-bit timer mode and sets the input clock to φT1
                                    (0.32 μs at fc = 25 MHz).
TREG1    ←      0 1 1 0 0 1 0 0     Sets 32μs ÷ φT1 = 100 (64H) in the timer register.
INTET01  ←      1 1 0 1 - - - -     Sets INTT1 to level 5.
T8RUN    ←      - - - - - - 1 -     Starts timer 1 counting.
T16RUN   ←      1 X - - X X X X
```

Note : X ; Don't care - ; No change

For input clock selection, see the following table.

Table 3.8 (1) Selecting Interrupt Interval and the Input Clock Using 8-Bit Timer

Input Clock	Interrupt Interval (at fc = 25 MHz)	Resolution
φT1 (8/fc)	0.32 μs to 81.92 μs	0.32 μs
φT4 (32/fc)	1.28 μs to 327.7 μs	1.28 μs
φT16 (128/fc)	5.12 μs to 1.311 ms	5.12 μs
φT256 (2048/fc)	81.92 μs to 20.97 ms	81.92 μs

② Generating a square wave with a 50%-duty cycle

Invert the timer flip-flop at fixed intervals and output the timer flip-flop values to the timer output pin (TO1).

Example : To output a square wave from pin TO1 with an interval of 1.92 μs at fc = 25 MHz, set the registers in the following order. Use either timer 0 or 1. The example shows the register settings for timer 1.

		MSB 7	6	5	4	3	2	1	LSB 0	
T8RUN	←	–	X	–	–	–	–	0	–	Stops timer 1 and clears it to 0.
T01MOD	←	0	0	X	X	0	1	–	–	Sets 8-bit timer mode and sets the input clock to φT1.
TREG1	←	0	0	0	0	0	0	1	1	Sets 1.92 μs ÷ φT1 ÷ 2 = 3 in the timer register.
T02FFCR	←	–	–	–	–	1	0	1	1	Clears TFF1 to 0 and sets it to invert on a match detect signal from timer 1.
P9CR	←	X	X	X	X	–	–	1	–	Sets P91 to TO1.
P9FC	←	X	X	X	X	–	X	1	X	
T8RUN	←	–	–	–	–	–	–	1	–	Starts timer 1 counting.
T16RUN	←	1	X	–	–	X	X	X	X	

Note: X; Don't care – ; No change

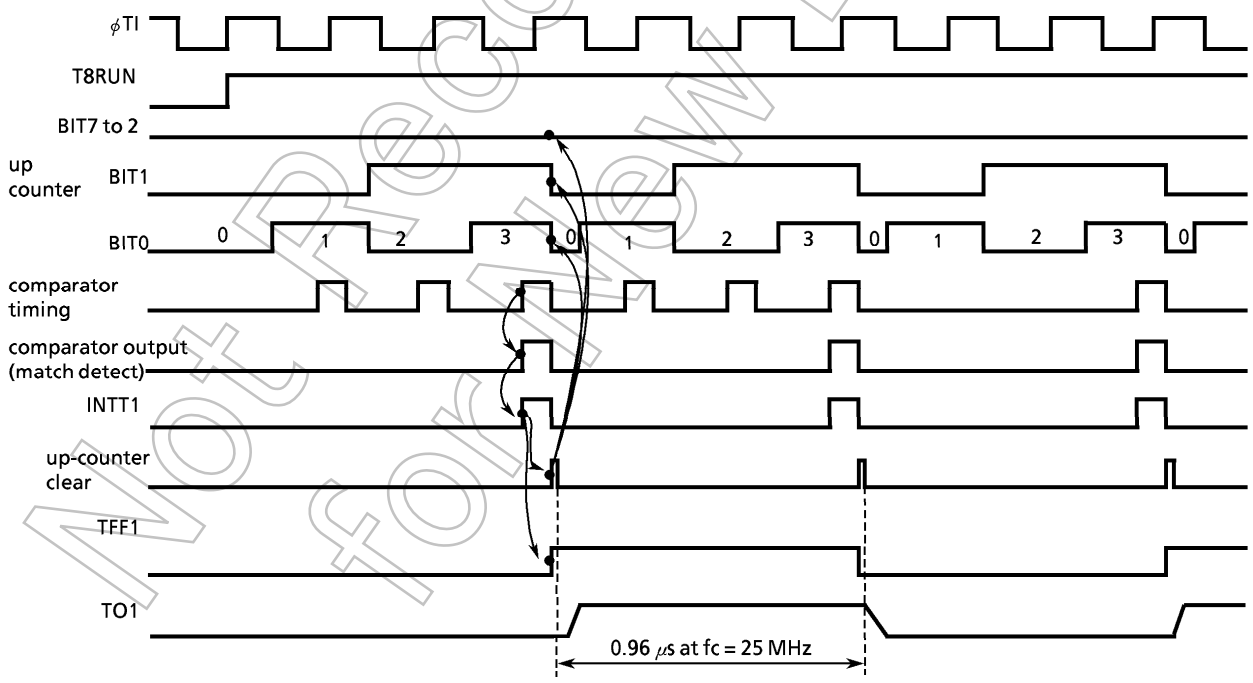


Figure 3.8 (13) Square Wave (50% Duty) Output Timing Chart

③ Setting timer 1 to count up at timer 0 match output

Set 8-bit timer mode and set the timer 1 input clock to timer 0 comparator output.

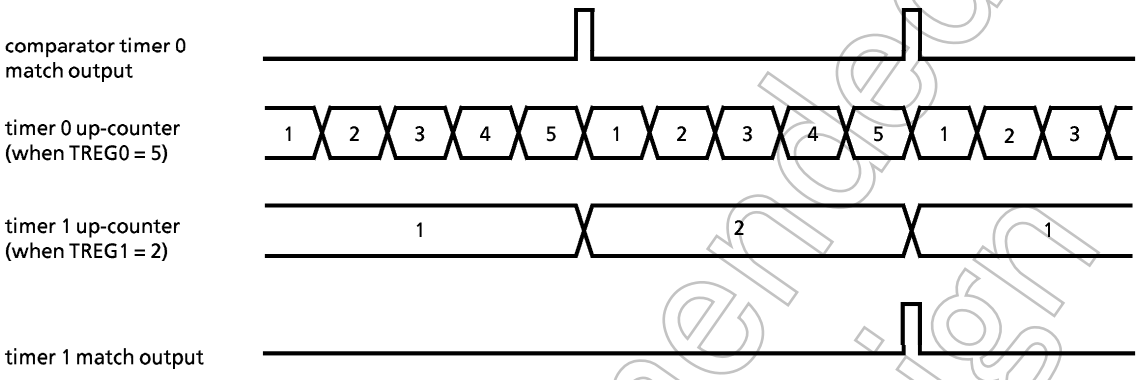


Figure 3.8 (14)

④ Output invert by software

The timer flip-flop (timer F/F) value can be inverted independently of timer operation.

For example, writing "00" to T02FFCR<FF1C1, 0> inverts the TFF1 value; writing "00" to T02FFCR<FF3C1, 0> inverts the TFF3 value.

⑤ Timer flip-flop (timer F/F) initialization

The timer flip-flop value can be initialized to "0" or "1" independently of timer operation.

For example, to set TFF1 to 0, write "10" to T02FFCR<FF1C1, 0>. To set TFF1 to 1, write "01" to T02FFCR<FF3C1, 0>.

Note : The timer flip-flop or timer register value cannot be read.

(2) 16-Bit Timer Mode

Timers 0 and 1, 2 and 3, 4 and 5, or 6 and 7 can be paired to configure 16-bit interval timers.

As timers 0 and 1, 2 and 3, 4 and 5, and 6 and 7 operate the same, the following describes timers 0 and 1 only.

To cascade-connect timers 0 and 1 and configure a 16-bit interval timer, set mode register T01MOD<T01M1, 0> to "01".

When setting 16-bit timer mode, the input clock for timer 1 is provided by the overflow output of timer 0, irrespective of the clock control register TCLK setting.

Table 3.8 (2) Selection of 16-Bit Timer (Interrupt) Interval and Input Clock

Input Clock	Interrupt Interval (fc = 25 MHz)	Resolution
ϕT1 (8/fc)	0.32 μs to 20.971 ms	0.32 μs
ϕT4 (32/fc)	1.28 μs to 83.885 ms	1.28 μs
ϕT16 (128/fc)	5.12 μs to 335.539 ms	5.12 μs

To set the timer interrupt interval, set the lower eight bits in timer register TREG0 and the upper eight bits in TREG1. Be sure to set TREG0 first (as entering data in TREG0 temporarily disables the compare, while entering data in TREG1 starts the compare).

Setting Example : To generate interrupt INTT1 every 0.32 s at fc = 25 MHz, set the following values in timer registers TREG0 and TREG1:
Using ϕT16 (= 5.12 μs at 25 MHz) as a timer input clock,
0.32 s ÷ 5.12 μs = 62500 = F424H
Therefore, set TREG1 to F4H, and TREG0 to 24H.

A match between up-counter UC0 and TREG0 triggers the timer 0 comparator to generate a match detect signal, but does not clear up-counter UC0. No interrupt INTT0 is generated.

A match between up-counter UC1 and TREG1 at comparator timing triggers the timer 1 comparator to generate a match detect signal. When comparator match detect signals for both timer 0 and timer 1 are generated, up-counter 0 and up-counter 1 are cleared to 0 and interrupt INTT1 only is generated. When invert is enabled, the value of timer flip-flop TFF1 is inverted.

	Timer 0			Timer 1		
	INT T0	TO1	Match Value	INT T1	TO1	Match Value
16-bit timer mode (timer 1 counts up on timer 0 overflow)	no interrupt generated	output disabled	TREG0 (timer 1 continues counting up at match)	interrupt generated	output enabled	TREG1*2^8 + TREG0 (full 16 bits)
8-bit timer mode (timer 1 counts up on timer 0 match)	interrupt generated	output enabled (timer 0 or timer 1)	TREG0 clear at match	interrupt generated	output enabled (timer 0 or timer 1)	TREG1* TREG0 (product)

Example : When TREG1 = 04H, and TREG0 = 80H :

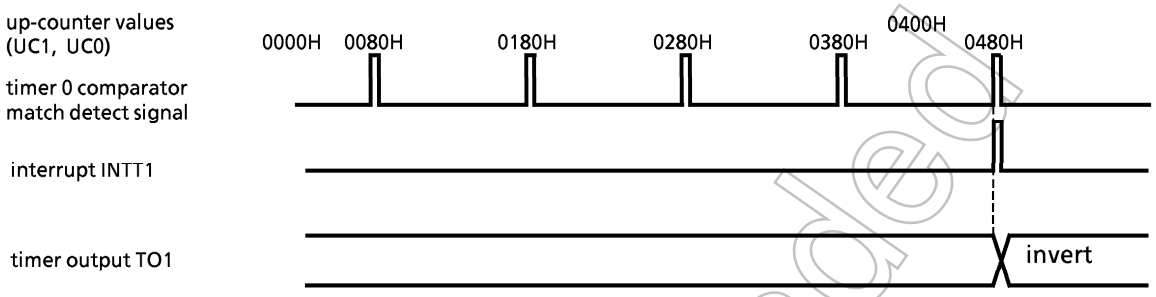


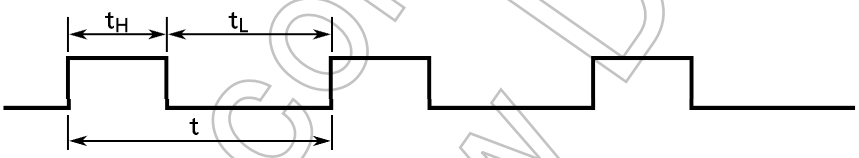
Figure 3.8 (15) Timer Output for 16-Bit Timer Mode

(3) 8-Bit Programmable Pulse Generation Output Mode

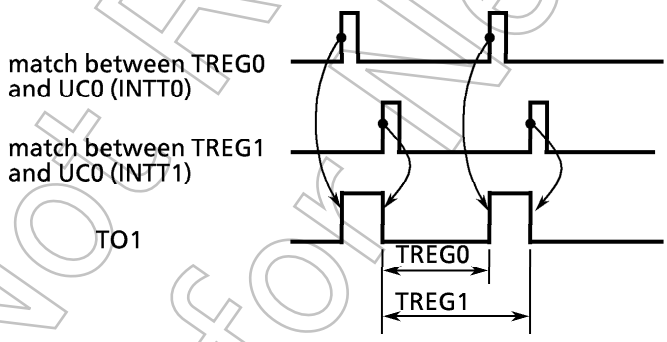
Timers 0, 2, 4, or 6 can output variable frequencies and square waves (pulses) with variable duty. The output pulse can be set to either active low or active high.

Timers 1, 3, 5, and 7 cannot be used in this mode.

Timer 0 outputs from pin TO1 (also used as P91), timer 2 outputs from pin TO3 (also used as P93), timer 4 outputs from TO5 (also used as P95), and timer 6 outputs from TO7 (also used as P97).



As timers 0, 2, 4, and 6 operate the same, the following describes timer 0 only.



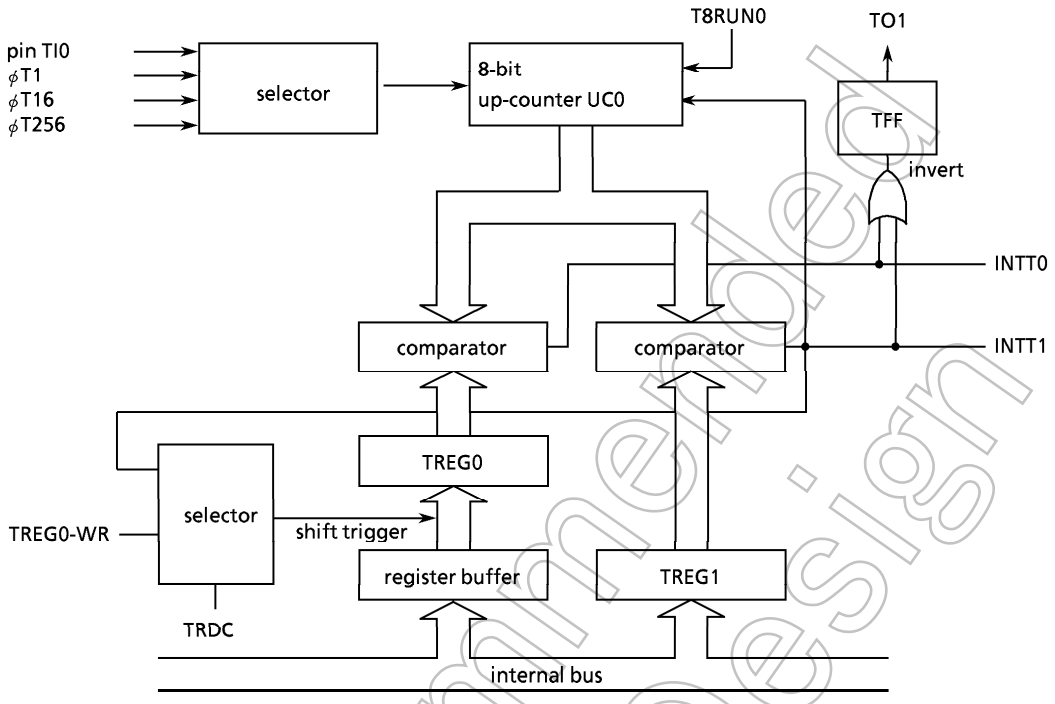
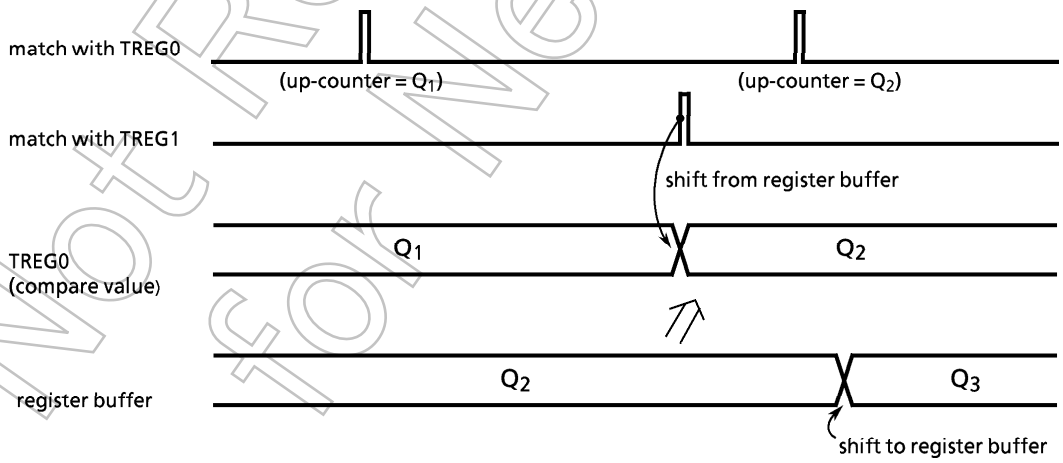


Figure 3.8 (16) 8-Bit PPG Output Mode Block Diagram

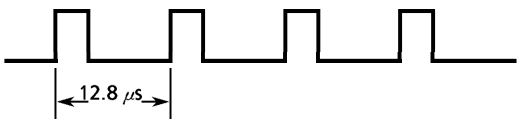
Enabling the TREG0 double-buffer in this mode shifts the register buffer value to TREG0 when TREG1 matches UC0.

Using the double-buffer facilitates output of waveforms with a low duty ratio (when changing the duty).



Register Buffer Operation

Example : Output a 1/4-duty 78.125 kHz-pulse (at fc = 25 MHz)



- Determine the set value in the timer register.
  Setting the frequency to 78.125 kHz generates a square wave with a cycle of t = 1/78.125 kHz = 12.8 μs.
  Using φT1 = 0.32 μs (at fc = 25 MHz) results in:
  12.8 μs ÷ 0.32 μs = 40
  Accordingly, set timer register 1 (TREG1) to TREG1 = 40 = 28H.
  Next, set the duty to 1/4 as follows: t × 1/4 = 12.8 μs × 1/4 = 3.2 μs
  Accordingly, set timer register 0 (TREG0) to TREG0 = 10 = 0AH.

```
          MSB          LSB
        ← 7 6 5 4 3 2 1 0
T8RUN   ← - - - - - - 0 0   Stops timers 0 and 1, and clear them to 0.
T16RUN  ← 0 X - - X X X X
T01MOD  ← 1 0 X X 0 1 0 1   Sets 8-bit PPG mode and sets the input clock to φT1.
T02FFCR ← - - - - 0 1 1 x   Sets TFF1 and enables invert.
                            Setting to "10" obtains a negative logic output wave.
TREG0   ← 0 0 0 0 1 0 1 0   Writes 0AH.
TREG1   ← 0 0 1 0 1 0 0 0   Writes 28H.
P9CR    ← X X X X - - 1 -   Sets P91 to TO1.
P9FC    ← X X X X - - 1 X
T8RUN   ← - - - - - - 1 1   Starts timers 0 and 1 counting.
T16RUN  ← 1 X - - X X X X
```

Note : X ; Don't care - ; No change

(4) 8-Bit Pulse Width Modulation (PWM) Output Mode

Only timers 0, 2, 4, and 6 support this mode, which allows up to two pulse width modulation outputs with 8-bit resolution.

For timer 0, PWM is output to pin TO1 (also used as P91). For timers 2, 4, and 6, PWM is output to pins TO3 (also used as P93), TO5 (also used as P95), and TO7 (also used as P97) respectively.

Timers 1, 3, 5, and 7 can be used as 8-bit timers.

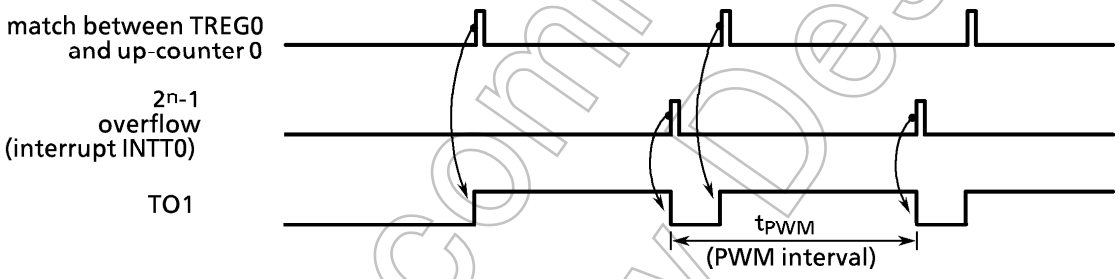
As timers 0, 2, 4, and 6 operate the same, the following describes timer 0 only.

Timer output is inverted when the up-counter UC0 setting and the timer register TREG setting match, or when 2^n-1 (T01MOD specifies one of n=6, n=7, or n=8) counter overflow occurs. The up-counter UC0 is cleared by the 2^n-1 counter overflow.

In 8-bit PWM output mode, the following conditions must be satisfied :

(timer register setting) $<$ (2^n-1 counter overflow settin)

(timer register setting) $\neq$ 0



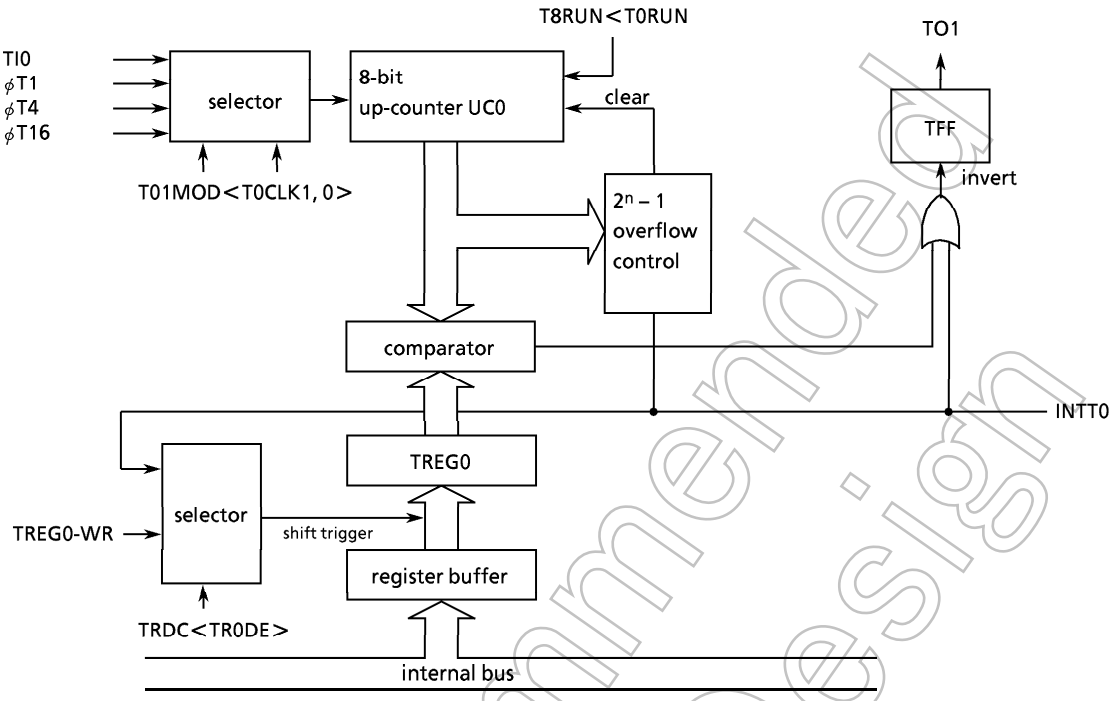
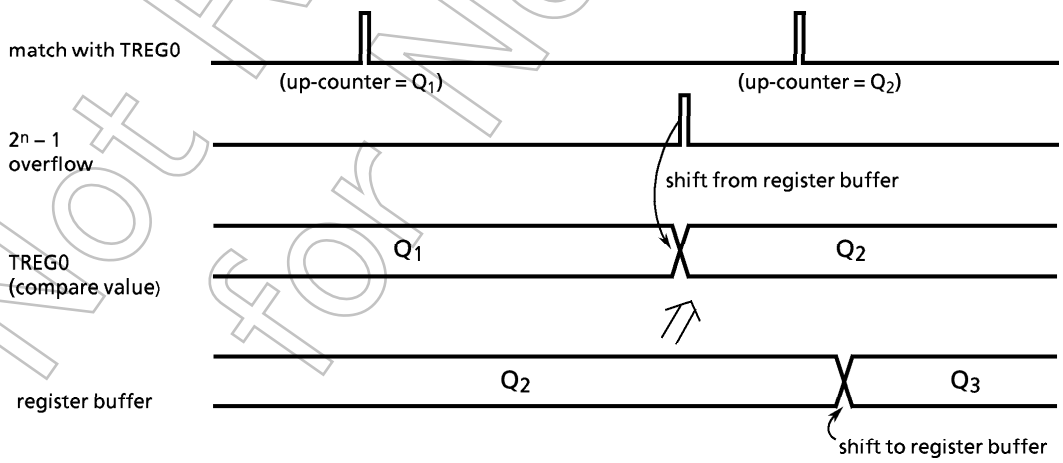


Figure 3.8 (17) 8-Bit PWM Output Mode Block Diagram

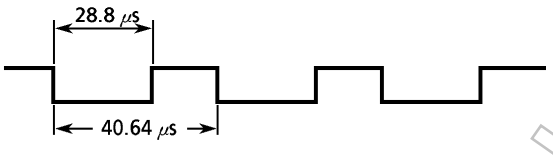
Enabling the TREG0 double-buffer in this mode shifts the register buffer value to TREG0 when 2^n-1 overflow is detected.

Using the double-buffer facilitates output of waveforms with a low duty ratio (when changing the duty).



Register Buffer Operation

Example : Output the following PWM waveform to pin TO1 using timer 0 for fc = 25 MHz:



To realize a PWM interval of 40.64 μs using φT1 = 0.32 μs (at fc = 25 MHz)

$40.64\ \mu s \div 0.32\ \mu s = 127 = 2^n - 1$

Accordingly, set n = 7.

As the low-level interval is 28.8 μs, at φT1 = 0.32 μs,

set 28.8 μs ÷ 0.32 μs = 90 = 5AH in TREG0

	MSB 7 6 5 4 3 2 1 0 LSB	
T8RUN	← - - - - - - - 0	Stops timer 0 and clears it to 0.
T01MOD	← 1 1 1 0 - - 0 1	Sets to 8-bit PWM mode (interval = $2^7 - 1$) and sets the input clock to φT1.
T02FFCR	← - - - - 1 0 1 X	Clears TFF1 and sets to invert enable.
TREG0	← 0 1 0 1 1 0 1 0	Writes 5AH.
P9CR	← X X X X - - 1 -	Sets P91 to TO1.
P9FC	← X X X X - - 1 X	
T8RUN	← - - - - - - - 1	Starts timer 0 counting.
T16RUN	← 1 X - - X X X X	

Note : X ; Don't care - ; No change

Table 3.8 (3) Setting PWM Interval and $2^n - 1$ Counter

	PWM Interval (at fc = 25 MHz)		
	φT1	φT4	φT16
$2^6 - 1$	20.2 μs (49.6 kHz)	80.6 μs (12.4 kHz)	322.6 μs (3.1 kHz)
$2^7 - 1$	40.6 μs (24.6 kHz)	162.6 μs (6.2 kHz)	650.2 μs (1.5 kHz)
$2^8 - 1$	81.6 μs (12.3 kHz)	326.4 μs (3.1 kHz)	1.31 ms (0.8 kHz)

(5) Table 3.8 (4) shows the settings for all 8-bit timer modes.

Table 3.8 (4) Setting Register for All Timer Modes

Timer Mode (for 8-bit timer x 2 channels)	Mode T01M (T23M) (T45M) (T67M)	PWM0 (PWM2) (PWM4) (PWM6)	Upper Timer Input Clock T1CLK (T3CLK) (T5CLK) (T7CLK)	Lower Timer Input Clock T0CLK (T2CLK) (T4CLK) (T6CLK)	Invert Select FF1IS (FF3IS) (FF5IS) (FF7IS)
16-bit timer (full 16 bits) × 1ch	01	–	–	(external, φT1, 4, 16)	–
8-bit timer (8-bit × 8-bit mode × 1ch) (inputs lower timer comparator output to upper timer)	00	–	00	(external, φT1, 4, 16)	0: lower timer 1: upper timer
8-bit timer × 2ch	00	–	(φT1, 16, 256)	(external, φT1, 4, 16)	0: lower timer 1: upper timer
8-bit PPG × 1ch	10	–	–	(external, φT1, 4, 16)	–
8-bit PWM × 1ch (lower) 8-bit timer × 1ch (upper)	11	PWM interval	(φT1, 16, 256)	(external, φT1, 4, 16)	–

3.9 16-Bit Timers

TMP95C063 incorporates two multi-function 16-bit timer/event counters (timers 8 and 9).

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) output mode
- Frequency measurement mode
- Pulse width modulation (PWM) mode
- Time differential measurement mode

The timer/event counters have a 16-bit up-counter, two 16-bit timer registers (one with a double-buffer configuration), two 16-bit capture registers, two comparators, capture input control, and timer flip-flops and accompanying F/F control circuit.

The timer/event counter is controlled by four control registers: T8MOD/T9MOD, T8FFCR/T9FFCR, T16RUN, and T89CR.

Figure 3.9 (1) is a block diagram of a 16-bit timer/event counter (timer 8). Timer 9 has the same circuit configuration.

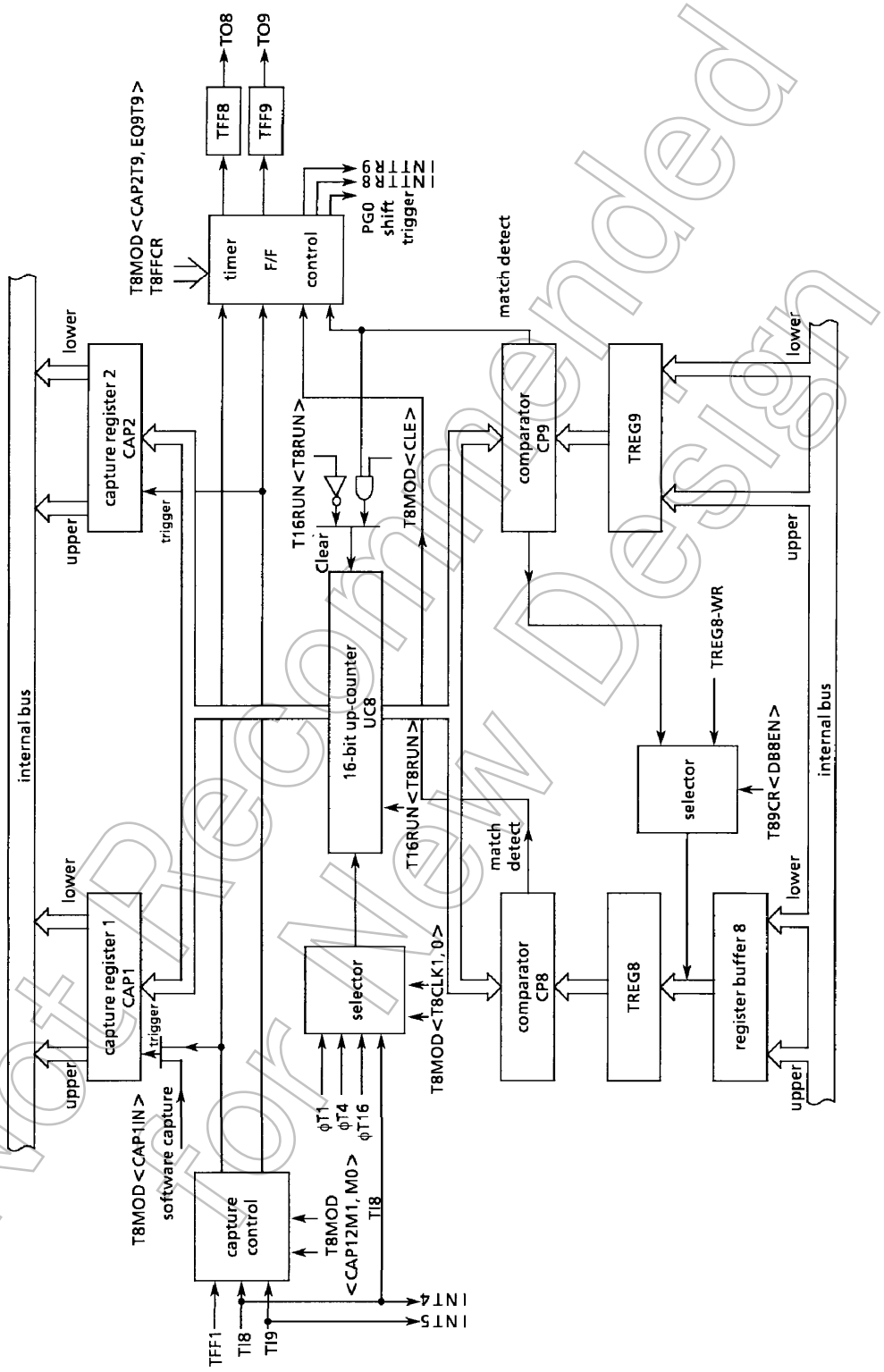


Figure 3.9 (1) 16-Bit Timer Block Diagram (Timer 8)

T8MOD (0038H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T9	EQ9T9	CAP1IN	CAP12M1	CAP12M0	CLE	T8CLK1	T8CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFF9 invert trigger 0: Disable trigger 1: Enable trigger At loading of up-counter value to CAP2 / At match between up-counter and TREG9		0: Soft-Capture 1: don't care	Capture timing 00: Disable INT4 at rising edge 01: TI8↑ TI9↑ INT4 at rising edge 10: TI8↑ TI8↓ INT4 at falling edge 11: TFF1↑ TFF1↓ INT4 at rising edge		1: UC8 Clear Enable	Timer 8 source clock 00: TI8 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 8 input clock

00	External input clock(TI8)
01	ϕT1 (8/fc)
10	ϕT4 (32/fc)
11	ϕT16 (128/fc)

Up-counter UC8 clear

0	Disables up-counter clear.
1	Clears by match with TREG9.

Figure 3.9 (2) 16-Bit Timer Mode Control Register (T8MOD) (1/2)

T8MOD (0038H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T9	EQ9T9	CAP1IN	CAP12M1	CAP12M0	CLE	T8CLK1	T8CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFF9 invert trigger 0: Disable trigger 1: Enable trigger		0: Soft-Capture 1: don't care	Capture timing 00: Disable INT4 at rising edge 01: TI8↑ TI9↑ INT4 at rising edge 10: TI8↑ TI8↓ INT4 at falling edge 11: TFF1↑ TFF1↓ INT4 at rising edge		1: UC8 Clear Enable	Timer 8 source clock 00: TI8 01: ϕT1 10: ϕT4 11: ϕT16	
	At loading of up-counter value to CAP2	At match between up-counter and TREG9						

Timer 8 capture timing

	Capture control	INT8 control
00	Capture disable	
01	CAP1 on TI8 rising CAP2 on TI9 rising	INT4 generated on TI8 rising
10	CAP1 on TI8 rising CAP2 on TI8 falling	INT4 generated on TI8 falling
11	CAP1 on TFF1 rising CAP2 on TFF1 falling	INT4 generated on TI8 rising

Software capture

0	Loads up-counter 8 value to CAP1.
1	Always read as "1".

Timer flip-flop 9 invert trigger (TFF9)

0	Trigger disable (invert disabled)
1	Trigger enable (invert enabled)

CAP2T9 : At loading of up-counter value to CAP2
EQ9T9 : At match between up-counter and TREG9

Figure 3.9 (3) 16-Bit Timer Control Register (T8MOD) (2/2)

T8FFCR (0039H)

	7	6	5	4	3	2	1	0
bit Symbol	TFF9C1	TFF9C0	CAP2T8	CAP1T8	EQ9T8	EQ8T8	TFF8C1	TFF8C0
Read/Write	W		R/W	R/W	R/W	R/W	W	
After reset	–		0	0	0	0	–	
Function	00: Invert TFF9 01: Set TFF9 10: Clear TFF9 11: don't care *Always read as "11"		TFF8 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF8 01: Set TFF8 10: Clear TFF8 11: don't care *Always read as "11"	
			At loading of up-counter value to CAP2	At loading of up-counter value to CAP1	At match between up-counter and TREG9	At match between up-counter and TREG8		

Timer flip-flop 8 (TFF8) control

00	Inverts TFF8 value (software invert).
01	Sets TFF8 to "1".
10	Clears TFF8 to "0".
11	Don't care (Always read as "11")

Timer flip-flop 8 (TFF8) invert trigger

0	Trigger disable (invert disabled)
1	Trigger enable (invert enabled)

CAP2T8 : At loading of up-counter value to CAP2
CAP1T8 : At loading of up-counter value to CAP1
EQ9T8 : At match between up-counter and TREG9
EQ8T8 : At match between up-counter and TREG8

Timer flip-flop 9 (TFF9) control

00	Inverts TFF9 value (software invert).
01	Sets TFF9 to "1".
10	Clear TFF9 to "0".
11	Don't care (Always read as "11")

Figure 3.9 (4) 16-Bit Timer 8 F/F Control (T8FFCR)

T9MOD (0048H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP4TB	EQBTB	CAP1IN	CAP12M1	CAP12M0	CLE	T9CLK1	T9CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFFB invert trigger 0: Disable trigger 1: Enable trigger		0: Soft-Capture 1: don't care	Capture timing 00: Disable INT6 at rising edge 01: TIA↑ TIB↑ INT6 at rising edge 10: TIA↑ TIA↓ INT6 at falling edge 11: TFF1↑ TFF1↓ INT6 at rising edge		1: UC9 Clear Enable	Timer 9 source clock 00: TIA 01: ϕT1 10: ϕT4 11: ϕT16	
	At loading of up-counter value to CAP4	At match between up-counter and TREGB						

Timer 9 input clock

00	External input clock (TIA)
01	ϕT1 (8/fc)
10	ϕT4 (32/fc)
11	ϕT16 (128/fc)

Up-counter UC9 clear

0	Disables up-counter clear.
1	Clears by match with TREGB.

Figure 3.9 (5) 16-Bit Timer Mode Control Register (T9MOD) (1/2)

T9MOD (0048H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP4TB	EQBTB	CAP3IN	CAP34M1	CAP34M0	CLE	T9CLK1	T9CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFFB invert trigger 0: Disable trigger 1: Enable trigger		0: Soft-Capture 1: don't care	Capture timing 00: Disable INT6 at rising edge 01: TIA↑ TIB↑ INT6 at rising edge 10: TIA↑ TIA↓ INT6 at falling edge 11: TFF1↑ TFF1↓ INT6 at rising edge		1: UC9 Clear Enable	Timer 9 source clock 00: TIA 01: ϕT1 10: ϕT4 11: ϕT16	
	At loading of up-counter value to CAP4	At match between up-counter and TREGB						

Timer 9 capture timing

	Capture control	INTA control
00	Capture disable	INT6
01	CAP1 on TIA rising CAP2 on TIB rising	INT6 generated onTIA rising
10	CAP1 on TIA rising CAP2 on TIA falling	INT6 generated on TIA falling
11	CAP1 on TFF1 rising CAP2 on TFF1 falling	INT6 generated on TIA rising

Software capture

0	At loading of up-counter 9 value to CAP1
1	Always read as "1"

Invert trigger of timer flip-flop B (TFFB)

0	Trigger disable (invert disabled)
1	Trigger enable (invert enabled)

CAP4TB : At loading of up-counter value to CAP2
EQBTB : At match between up-counter and TREGB

Figure 3.9 (6) 16-Bit Timer Control Register (T9MOD) (2/2)

T9FFCR (0049H)

	7	6	5	4	3	2	1	0
bit Symbol	TFFBC1	TFFBC0	CAP4TA	CAP3TA	EQBTA	EQATA	TFFAC1	TFFAC0
Read/Write	W		R/W	R/W	R/W	R/W	W	
After reset	–		0	0	0	0	–	
Function	00: Invert TFFB 01: Set TFFB 10: Clear TFFB 11: don't care *Always read as "11"		TFFA invert trigger 0: Disable trigger 1: Enable trigger At loading of up-counter value to CAP4	At loading of up-counter value to CAP3	At match between up-counter and TREGB	At match between up-counter and TREGA	00: Invert TFFA 01: Set TFFA 10: Clear TFFA 11: don't care *Always read as "11"	

Timer flip-flop A (TFFA) control

00	Inverts TFFA value (software invert).
01	Sets TFFA to "1".
10	Clears TFFA to "0".
11	Don't care (Always read as "11")

Timer flip-flop A (TFFA) invert trigger

0	Trigger disable (invert disabled)
1	Trigger enable (invert enabled)

CAP4TA : At loading of up-counter value to CAP4
CAP3TA : At loading of up-counter value to CAP3
EQBTA : At match between up-counter and TREGB
EQATA : At match between up-counter and TREGA

Timer flip-flop B (TFFB) control

00	Inverts TFFB value (software invert).
01	Sets TFFB to 1.
10	Clears TFFB to 0.
11	Don't care (Always read as "11")

Figure 3.9 (7) 16-Bit Timer 9 F/F Control (T9FFCR)

T89CR (003AH)

	7	6	5	4	3	2	1	0
bit Symbol	—				PG1T	PG0T	DBAEN	DB8EN
Read/Write	R/W				R/W			
After reset	0				0	0	0	0
Function	Fixed to "0" * Always read as "11"				PG1 shift trigger 0: 8-bit timer trigger 3 (timers 2, 3) 1: 16-bit timer trigger (timer 9)	PG0 shift trigger 0: 8-bit timer trigger 1 (timers 0, 1) 1: 16-bit timer trigger (timer 8)	Double-buffer 0: Disable 1: Enable TREGA double-buffer	TREG8 double-buffer

Double-buffer control

0	Disable
1	Enable

DBAEN : TREGA double-buffer
DB8EN : TREG8 double-buffer

Figure 3.9 (8) 16-Bit Timer (8/9) Control Register (T89CR)

T16RUN (003BH)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T9RUN	T8RUN				
Read/Write	R/W		R/W	R/W				
After reset	0		0	0				
Function	Prescaler & Timer Run/Stop CONTROL 0: Stop & Clear 1: Run (Count up)							

16-bit timer (timer 8) operation

0	Stop and clear
1	Count up

16-bit timer (timer 9) operation

0	Stop and clear
1	Count up

Prescaler operation

0	Stop and clear
1	Count up

Figure 3.9 (9) 16-Bit Timer Operation Control Register (T16RUN)

① Up-counter

The up-counter is a 16-bit binary counter that counts up using the input clock specified by 16-bit timer mode control registers T8MOD<T8CLK1, 0> and T9MOD<T9CLK1, 0>.

The input clock is selected from internal clocks ϕT1, ϕT4, and ϕT16 output from the 9-bit prescaler (shared with the 8-bit timers), or the external clocks output from pin TI8 (also used as PB0/INT4) and pin TIA (also used as PB4/INT6). A reset initializes <T8CLK1, 0>/ <T9CLK1, 0> to "00", selecting an external input clock on pin TI8/TIA as the input clock.

To control the count, stop, and clear functions for the counter, use timer control register T16RUN<T8RUN, T9RUN>.

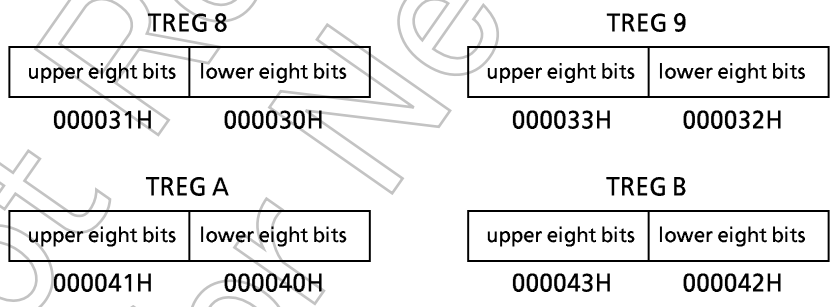
If up-counter clearing is enabled, up-counter UC8/9 is cleared to 0 when up-counter UC8/9 matches timer register TREG9/B. The clear enable/disable is set with T8MOD<CLE> and T9MOD<CLE>.

When clear disable is set, the counter operates as a free-running counter.

② Timer registers

Each timer has two internal 16-bit registers for setting counter values. When the value set in the timer register matches the value of the up-counter UC8/9, the comparator match detect signal is activated.

Data set to timer register TREG8, TREG9/A, or TREGB uses the 2-byte data load instruction, or the 1-byte data load instruction twice; first to write data to the lower eight bits, then to write data to the upper eight bits.



Timer registers TREG8 and TREGA have a double-buffer configuration and are paired with a register buffer. Timer registers TREG8/A enable/disable the double-buffer function using timer control register T89CR<DB8EN, DBAEN>. Setting <DB8EN, DBAEN> to 0 disables the double-buffer; setting <DB8EN, DBAEN> to 1 enables the double-buffer.

With the double-buffer enabled, data are transmitted from the register buffer to the timer register at a match between up-counter UC8/9 and timer register TREG9/B.

A reset initializes T89CR<DB8EN, DBAEN> to "0", disabling the double-buffer. When using the double-buffer, write data to the timer register and set <DB8EN, DBAEN> to "1", then write the next data to the register buffer.

TREG8/TREGA and the register buffer are allocated to the same addresses in memory (000030H, 000031H/000040H, 000041H).

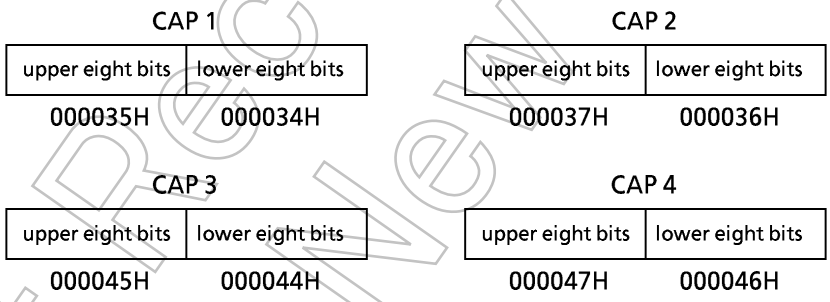
When <DB8EN, DBAEN> is set to "0", the same value is written to TREG8 and TREGA and to their respective register buffers. When <DB8EN, DBAEN> is set to "1", the value is written to the register buffers only. Therefore, disable the register buffers before writing the initial values to the timer registers.

As the timer registers are undefined after a reset, be sure to write data to the upper and lower registers before using the timers.

③ Capture register

The capture register is a 16-bit register for latching the up-counter value.

When reading the capture register, use the 2-byte data load instruction, or the 1-byte data load instruction twice; first to read data from the lower eight bits, then to read data from the upper eight bits.



④ Capture input control

The capture input control circuit controls the timing to latch the up-counter UC8/9 value to capture registers CAP1, CAP2/3, and CAP4.

Set the capture register latch timing using T8MOD<CAP12M1, 0> / T9MOD<CAP34M1, 0>.

- When T8MOD<CAP12M1,0>/T9MOD<CAP34M1,0> = "00",

  the capture function is disabled. Resetting disables the capture function.

- When T8MOD<CAP12M1, 0>/T9MOD<CAP34M1, 0> = "01"

  On the TI8 (also used as PB0/INT4) /TIA (also used as PB4/INT6) input rising edge, the up-counter value is loaded to capture register CAP1/CAP3. On the TI9 (also used as PB1/INT5)/TIB (also used as PB5/INT7) input rising edge, the up-counter value is loaded to capture register CAP2/CAP4. (Time differential measurement)

- When T8MOD<CAP12M1, 0>/T9MOD<CAP34M1, 0> = "10"

  On the TI8/TIA input rising edge, the up-counter value is loaded to capture register CAP1/CAP3. On the input falling edge, the up-counter value is loaded to capture register CAP2/CAP4. In this mode only, interrupt INT4/6 is generated on a falling edge. (Pulse width measurement)

- When T8MOD<CAP12M1, 0>/T9MOD<CAP34M1, 0> = "11"

  On the timer flip-flop TFF1 rising edge, the up-counter value is loaded to capture register CAP1/CAP3. On the falling edge, the up-counter value is loaded to capture register CAP2/CAP4.

  The up-counter value can also be loaded to a capture register on a software request. When "0" is written to T8MOD<CAP1IN>/T9MOD<CAP3IN>, the up-counter value at that time is loaded to capture register CAP1/3. The prescaler must be set to RUN (set T16RUN<PRRUN> to "1").

⑤ Comparator

A 16-bit comparator compares the up-counter UC8/UC9 value with the value set in the timer register (TREG8, TREG9/TREGA, or TREGB) to detect a match.

On detection of a match, the comparator generates interrupt INTTR8, INTTR9/INTRRA, or INTTRB.

Only a match with TREG9/B clears up-counter UC8/9. (Setting T8MOD<CLE>/T9MOD<CLE> to "0" disables UC8/9 clearing.)

⑥ Timer flip-flop TFF8/TFFA

This flip-flop is inverted by a match detect signal from the comparator and a latch signal to the capture register.

Enable or disable the invert for each interrupt source using T8FFCR<CAP2T8, CAP1T8, EQ9T8, EQ8T8>/T9FFCR<CAP4TA, CAP3TA, EQBTA, EQATA>.

To invert TFF8/A, write "00" to T8FFCR <TFF8C1, 0> / T9FFCR <TFFAC1, 0>. Writing "01" sets TFF8/A to 1; "10" clears TFF8/A to 0.

The TFF8/A value can be output to timer output pin TO8 (also used as PB2)/TOA (also used as PB6).

⑦ Timer flip-flop TFF9/TFFB

This flip-flop is inverted by a match detect signal between up-counter UC8/9 and timer register TREG9/B, and a latch signal to capture register CAP2/4.

Enable or disable the invert for each interrupt source using T8MOD<CAP2T9, EQ9T9>/T9MOD<CAP4TB, EQBTB>.

To invert TFF9/B, write "00" to T8FFCR<TFF9C1, 0> / T9FFCR<TFFBC1, 0>. Writing "01" sets TFF9/B to 1; "10" clears TFF9/B to 0.

The TFF9/B value can be output to timer output pin TO9 (also used as PB3)/ TOB (also used as PB7).

(1) 16-Bit Timer Mode

Timers 8 and 9 operate independently. As both timers operate the same, the following describes timer 8 only.

Example : Generate fixed-interval interrupts

Set an interval time in timer register TREG9 and generate interrupt INTTR9.

```
          7 6 5 4 3 2 1 0
T16RUN  ← - X - 0 X X X X   Stop timer 8.
INTET89 ← 1 1 0 0 1 0 0 0   Enables INTTR9 (set to level 4) and disables INTTR8.
T8FFCR  ← 1 1 0 0 0 0 1 1   Disables trigger.
T8MOD   ← 0 0 1 0 0 1 * *   Sets input clock to an internal clock, and disables
            (** = 01, 10, 11) capture function.
TREG9   ← * * * * * * * *   Sets interval time.
          * * * * * * * *   (16 bits)
T16RUN  ← 1 X - 1 X X X X   Starts timer 8.

Note: X ; Don't care      - ; No change
```

(2) 16-Bit Event Counter Mode

Setting external clock TI8/TIA as an input clock in 16-bit timer mode results in an event counter. To obtain a counter value, load the counter value into a capture register using "software capture" and read the captured value from the capture register.

The counter counts up at the TI8/TIA input rising edge.

The TI8/TIA pin is also used as PB0, INT4/PB4, INT6.

As timers 8 and 9 operate the same, the following describes timer 8 only.

```
                7 6 5 4 3 2 1 0
T16RUN  ← - X - 0 X X X X      Stop s timer 8.
PBCR    ← - - - - - - - 0      Sets PB0 to input mode.
INTET89 ← 1 1 0 0 1 0 0 0      Enables INTTR9 (level 4) and disables INTTR8.
T8FFCR  ← 1 1 0 0 0 0 1 1      Disables trigger.
T8MOD   ← 0 0 1 0 0 1 0 0      Sets input clock to TI8.
TREG9   ← * * * * * * * *      Sets the count (16 bits).
          * * * * * * * *
T16RUN  ← 1 X - 1 X X X X      Starts timer 8.
```

Note: Set the prescaler to RUN when using a 16-bit counter as an event counter.

(3) 16-Bit Programmable Pulse Generation (PPG) Output Mode

As timers 8 and 9 operate the same, the following describes timer 8 only.

To enter PPG mode, set the device to invert timer flip-flop TFF8 and output the TFF8 value from the TO8 pin (also used as PB2) at a match between up-counter UC8 and the TREG8/TREG9 register value.

The following condition must be satisfied: (TREG8 setting) < (TREG9 setting).

```
                7 6 5 4 3 2 1 0
T16RUN  ← - X - 0 X X X X      Stops timer 8.
TREG8   ← * * * * * * * *      Sets the duty. (16 bits)
          * * * * * * * *
TREG9   ← * * * * * * * *      Sets the interval. (16 bits)
          * * * * * * * *
T89CR   ← 0 X X X - - - 1      Enables TREG8 double-buffer.
                               (Duty/interval modified by interrupt INTTR9)
T8FFCR  ← 1 1 0 0 1 1 1 0      Sets TFF8 to invert at detection of a match with
                               TREG8 or TREG9. Sets TFF8 initial value to "0".
T8MOD   ← 0 0 1 0 0 1 * *      Sets the input clock to the internal clock, and disable
          (** = 01, 10, 11)    the capture function.
PBCR    ← - - - - - - 1 - -  } Allocates PB2 to TO8.
PBFC    ← X - X X - 1 X X    } Starts timer 8.
T16RUN  ← 1 X - 1 X X X X
```

Note : X ; Don't care - ; No change

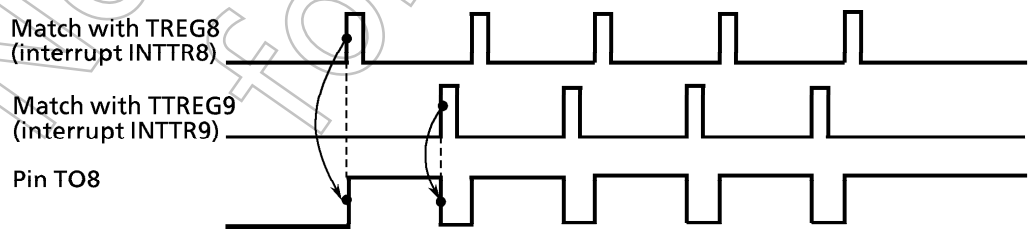


Figure 3.9 (10) Programmable Pulse Generation (PPG) Output Waveform

Enabling the TREG8 double-buffer in this mode shifts the value of register buffer 8 to TREG8 when TREG9 matches UC8. Using the double-buffer facilitates output of waveforms with a low duty ratio.

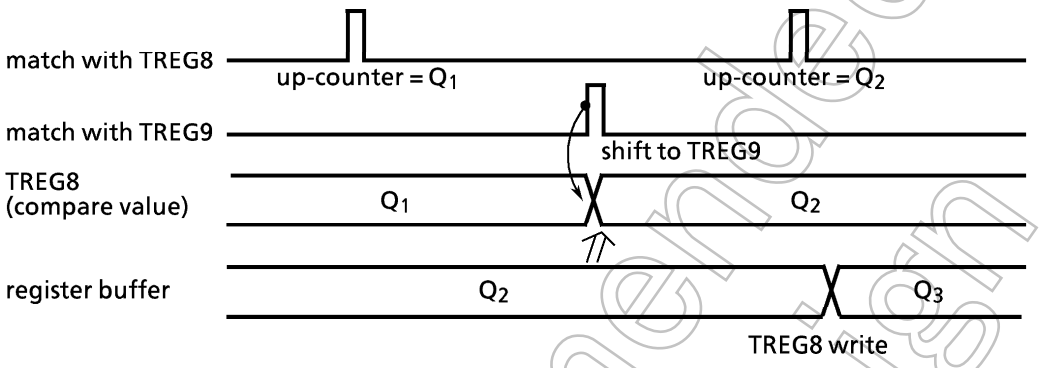


Figure 3.9 (11) Register Buffer Operation

The following is a block diagram of this mode.

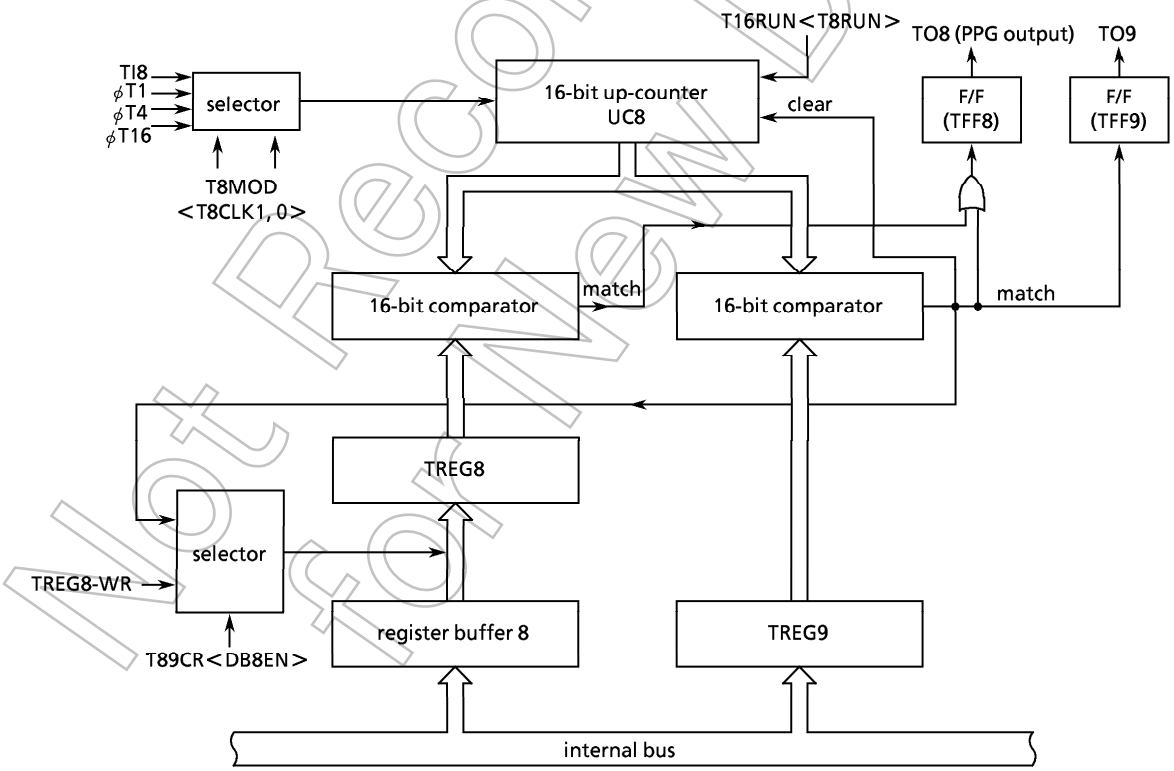


Figure 3.9 (12) 16-Bit PPG Mode Block Diagram

(4) Capture Function Application Example

As timers 8 and 9 operate the same, the following describes timer 8 only.

The following features of the 16-bit timer can be enabled or disabled as required: loading of up-counter UC8 value to capture registers CAP1 and CAP2, inversion of timer flip/flop TFF8 on a match detect signal from comparators CP8 and CP9, and outputting of TFF8 to pin TO8. Many functions can be obtained by combining these features with interrupts. For example:

① One-shot pulse output from the external trigger pulse

② Frequency measurement

③ Pulse width measurement

④ Time differential measurement

① One-shot pulse output from external trigger pulse

Set up-counter UC8 to free-running using internal clock input. Input the external trigger pulse from pin TI8, and load the up-counter value to capture register CAP1 on the TI8 input rising edge (set T8MOD<CAP12M1,0> to "01"). On the TI8 input rising edge, add the value of capture register CAP1 at interrupt INT4 (c) to the delay time (d), and set timer register TREG8 to the sum of these values (c + d). Add the pulse width of the one-shot pulse (p) to TREG8, and set TREG9 to the result (c + d + p). On interrupt INT4, set register T8FFCR<EQ9T8, EQ8T8> to "enable the inversion of timer flip-flop TFF8 only when the up-counter matches with TREG8 or TREG9". On interrupt INTTR9, disable the inversion of timer flip-flop TFF8.

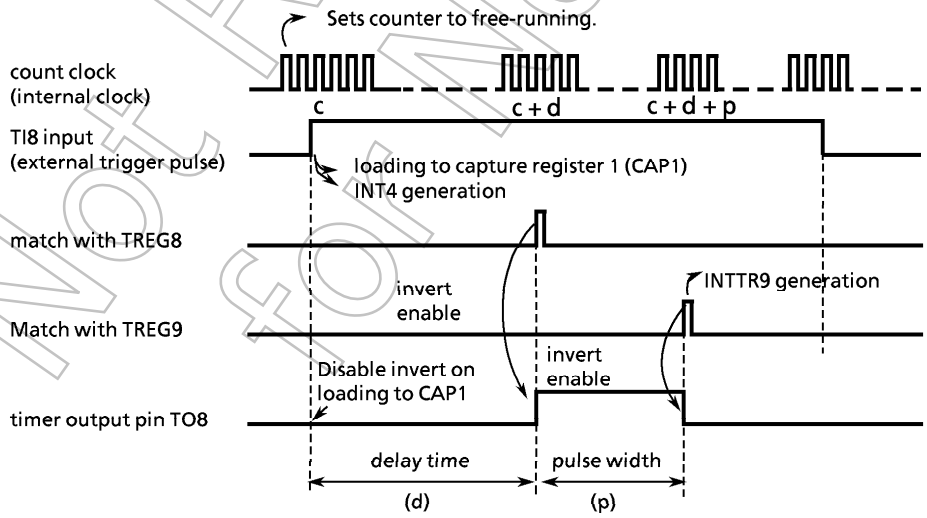
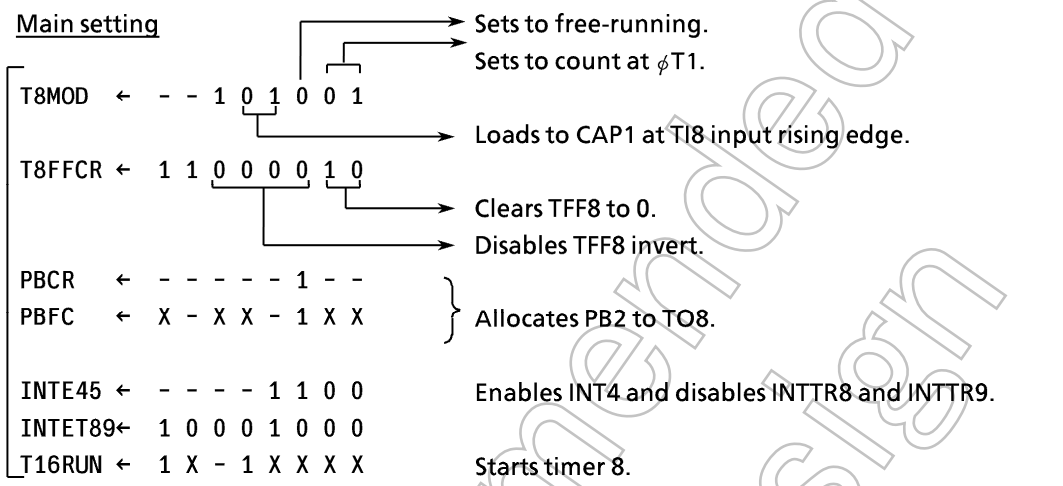
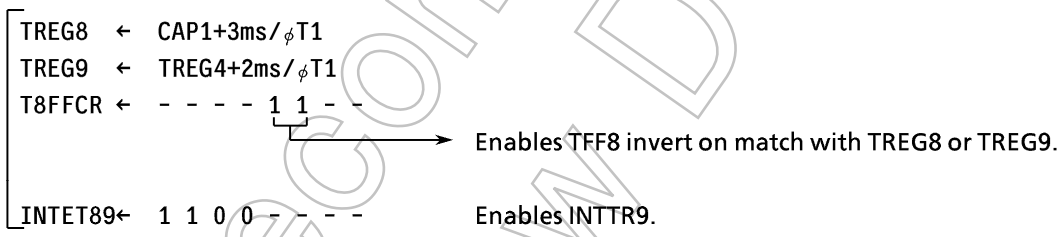
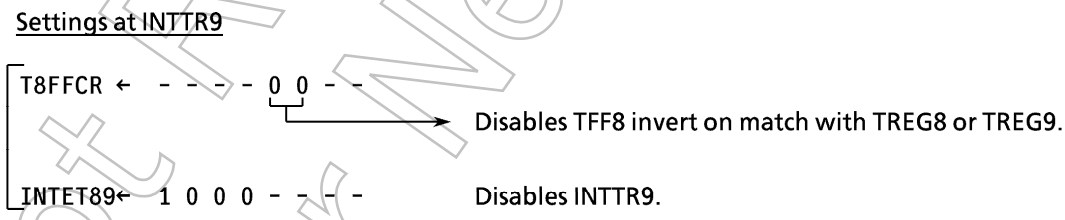


Figure 3.9 (13) One-Shot Pulse Output (With Delay)

Setting Example : On pin TI8, output a 2ms one-shot pulse with a 3ms-delay after an external trigger pulse.







Note: X ; Don't care - ; No change

If delay time is not required, invert timer flip-flop TFF8 by loading to capture register 1 (CAP1). Set timer register TREG9 to the sum of the one-shot pulse width (p) and the value of CAP1 at interrupt INT4 (c) (c + p). Enable TFF8 invert on match between TREG9 and up-counter UC8. On interrupt INTTR9, disable the timer flip-flop TFF8 invert.

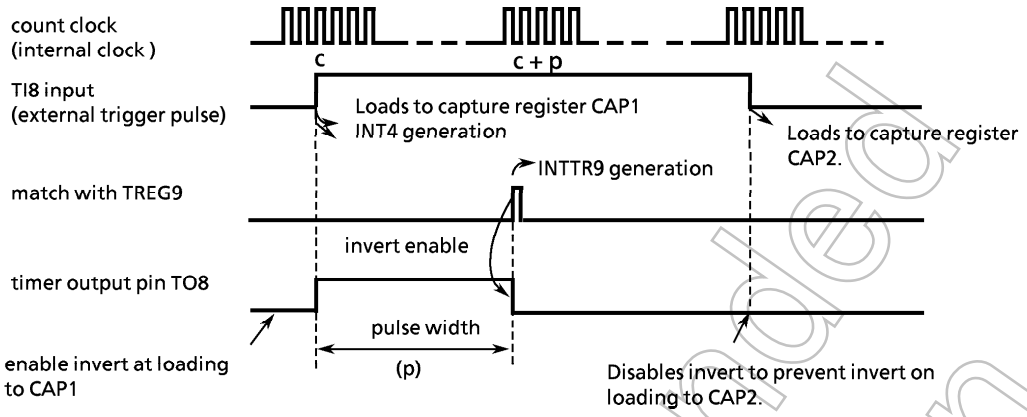


Figure 3.9 (14) One-Shot Pulse Output (No Delay)

② Frequency measurement

This mode is used to measure the frequency of the external clock. Input the external clock on pin TI8 and measure its frequency with the 8-bit timers (timers 0,1) and the 16-bit timer/event counter (timer 8).

Set the TI8 input as the timer 8 input clock, and load the value of up-counter UC8 to capture register CAP1 when timer flip/flop TFF1 of the 8-bit timer (timer 0,1) rises, and to capture register CAP2 when timer flip/flop TFF1 falls.

The frequency is determined from the difference between capture registers CAP1 and CAP2 at the 8-bit timer interrupts (INTT0 or INTT1).

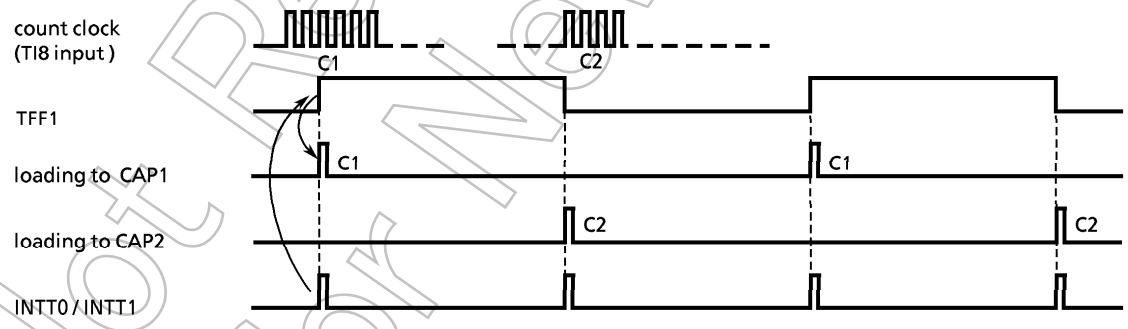


Figure 3.9 (15) Frequency Measurement

For example, if TFF1 is set to "1" for 0.5 s by the 8-bit timers, and the difference between CAP1 and CAP2 is 100, the frequency is 100 ÷ 0.5 [s] = 200 [Hz].

③ Pulse width measurement

This mode is used for measuring the "high" level width of an external pulse. Input the external pulse through pin TI8 and set the 16-bit timer/event counter to free-running count-up using an internal clock. Load the up-counter UC8 value into capture register CAP1 and CAP2 on the rising and falling edge respectively of the external pulse. Interrupt INT4 is generated on the falling edge of pin TI8.

The pulse width can now be determined according to the difference between CAP1 and CAP2, and the internal clock interval.

For example, if the difference between CAP1 and CAP2 is 100 and the internal clock interval is 0.8 μs, the pulse width is $100 \times 0.8\ \mu s = 80\ \mu s$.

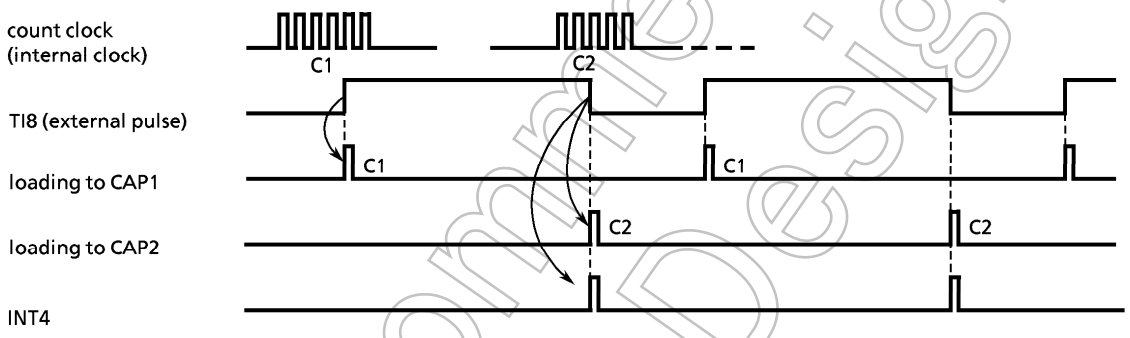


Figure 3.9 (16) Pulse Width Measurement

Note: Only in pulse width measurement mode where T8MOD<CAP12M1,0> = "10", external interrupt INT4 is generated at the falling edge of pin TI8. In other modes, external interrupt INT4 is generated at the rising edge.

Determine the "low" level width at the second INT4 using the difference between the value of C2 at the first interrupt and the value of C1 at the second interrupt.

④ Time differential measurement

This mode measures the time difference between the rising edge of the external pulses input to pins TI8 and TI9.

Set the 16-bit timer/event counter (timer 8) to free-running count-up using an internal clock. When a rising edge is detected in the pulse on pin TI8, the up-counter UC8 value is loaded into capture register CAP1 and interrupt INT4 is generated.

Similarly, when a rising edge is detected in the pulse on pin TI9, the up-counter UC8 value is loaded into capture register CAP2 and interrupt INT5 is generated.

When the up-counter values are loaded to CAP1 and CAP2, the time difference can be determined from the difference between CAP1 and CAP2.

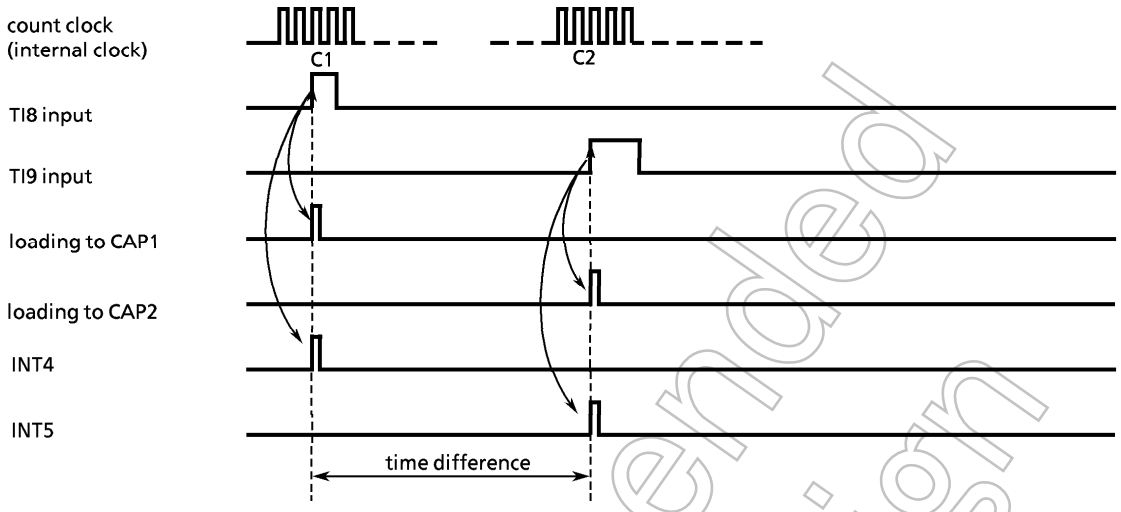


Figure 3.9 (17) Time Differential Measurement

(5) Phase Output Mode

Set the up-counter UC8/9 to free-running and output a signal with any phase differential. As timers 8 and 9 operate the same, the following describes timer 8 only.

A match between up-counter UC8 and TREG8 or TREG9 inverts TFF8 or TFF9 respectively, and outputs the invert values to TO8 and TO9 respectively.

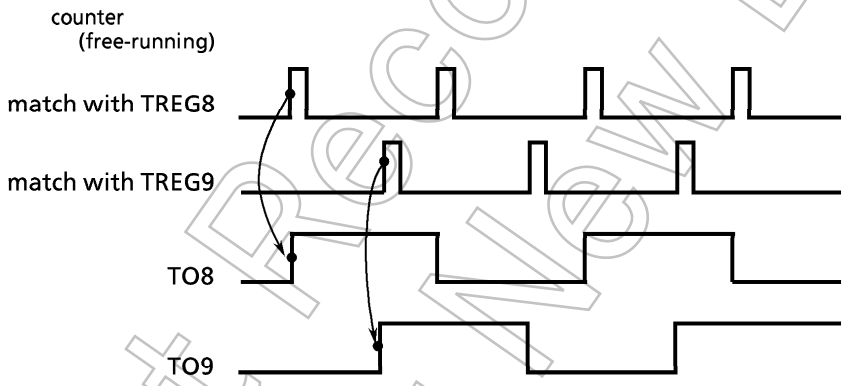


Figure 3.9 (18) Phase Output

The following table shows the interval (counter overflow time) of the above waveform output.

	20 MHz	25 MHz
ϕT1	26.214 ms	20.97 ms
ϕT4	104.856 ms	83.88 ms
ϕT16	419.424 ms	335.54 ms

3.10 Pattern Generator/Stepping Motor Control

TMP95C063 incorporates a 4-bit, 2-channel pattern generator/stepping motor control port (PG), linked with the 8-bit and 16-bit timers. PG shares pins with port 7 (an 8-bit I/O port). The two channels are PG0 and PG1.

The PG0 (channel 0) output is driven by 8-bit timers 0 and 1, or by 16-bit timer 8. The PG1 (channel 1) output is driven by 8-bit timers 2 and 3, or by 16-bit timer 9.

The PG01CR control register controls PG. Operation can be set to either pattern generation mode or stepping motor control mode.

The PG output shares pins with port 7, and any port 7 bit can be set for PG output.

Channel 0 (PG0) and channel 1 (PG1) operate independently.

As both channels operate identically apart from the differences shown below, the following describes channel 0 (PG0) only.

Difference between PG0 and PG1

	PG0	PG1
Timer trigger signal	From either 8-bit timer 0,1, or 16-bit timer 8	From either 8-bit timer 2,3, or 16-bit timer 9

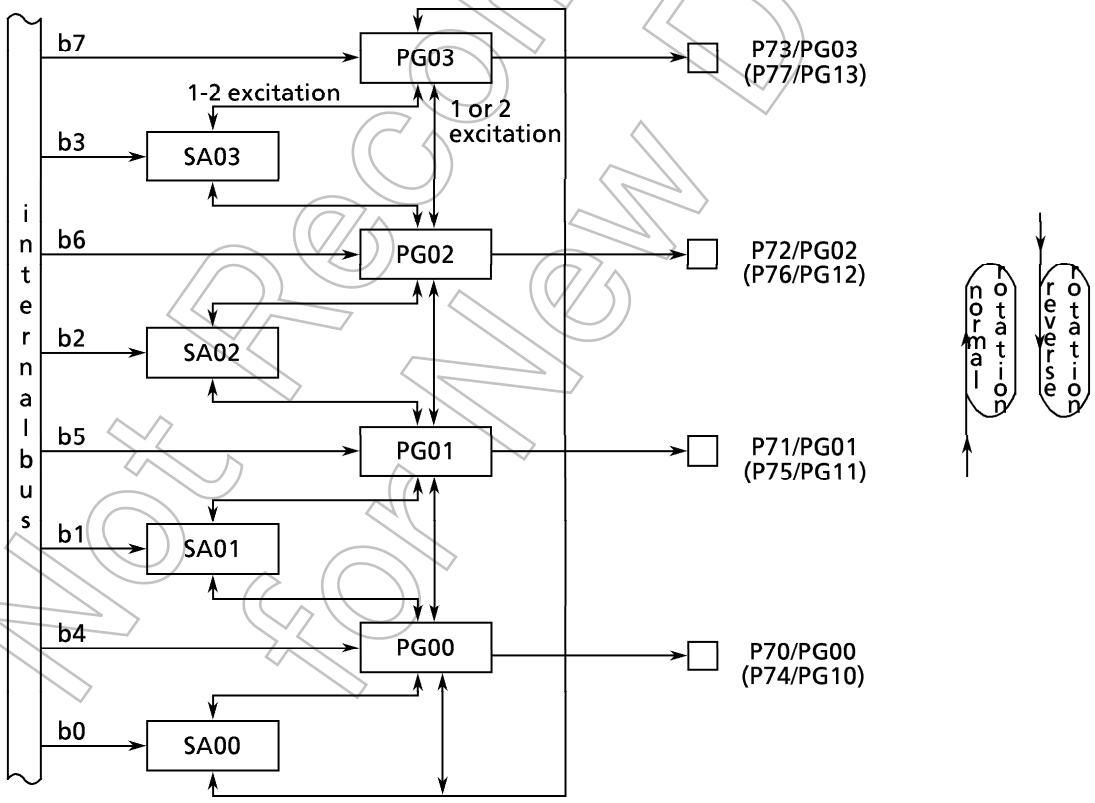


Figure 3.10 (1) Pattern Generator/Stepping Motor Control Block Diagram

PG01CR (004EH)

	7	6	5	4	3	2	1	0
bit Symbol	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
Read/Write	R / W				R / W			
After reset	0	0	0	0	0	0	0	0
Function	PG1 write mode 0: 8 bit write 1: 4 bit write	PG1 rotation direction 0: normal 1: reverse	PG1 mode (excitation) 0: either 1 or 2 excitation 1: 1-2 excitation	PG1 trigger input enable 0: disable 1: enable	PG0 write mode 0: 8 bit write 1: 4 bit write	PG0 rotation direction 0: normal 1: reverse	PG0 mode (excitation) 0: either 1 or 2 excitation 1: 1-2 excitation	PG0 trigger input enable 0: disable 1: enable

PG0 trigger input enable

0	Disables trigger input to PG0 block.
1	Enables trigger input to PG0 block.

PG0 operating mode setting

0	1 or 2 excitation (full-step)
1	1-2 excitation (half-step)/PG mode

PG0 (stepping motor control) rotation direction control

0	Normal / PG mode
1	Reverse

PG0 write mode setting

0	8-bit write
1	4-bit write/PG mode (write to shift alternate register only)

Figure 3.10 (2) (a) Pattern Generator Control Register (PG01CR)

PG01CR (004EH)

	7	6	5	4	3	2	1	0
bit Symbol	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
Read/Write	R / W				R / W			
After reset	0	0	0	0	0	0	0	0
Function	PG1 write mode 0: 8 bit write 1: 4 bit write	PG1 rotation direction 0: normal 1: reverse	PG1mode (excitation) 0: either 1 or 2 excitation 1: 1-2 excitation	PG1 trigger input enable 0: disable 1: enable	PG0 write mode 0: 8 bit write 1: 4 bit write	PG0 rotation direction 0: normal 1: reverse	PG0 mode (excitation) 0: either 1 or 2 excitation 1: 1-2 excitation	PG0 trigger input enable 0: disable 1: enable

PG1 trigger input enable

0	Disables trigger input to the PG1 block.
1	Enables trigger input to the PG1 block.

PG1 operating mode setting

0	1 or 2 excitation (full-step)
1	1-2 excitation (half-step)/PG mode

PG1 (stepping motor control) rotation direction control

0	Normal/PG mode
1	Reverse

PG1 write mode setting

0	8-bit write
1	4-bit write/PG mode (write to shift alternate register only)

Figure 3.10 (2) (b) Pattern Generator Control Register (PG01CR)

PG0REG (004CH)

	7	6	5	4	3	2	1	0
bit Symbol	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
Read/Write	W				R / W			
After reset	0	0	0	0	Undefined			
Function	Pattern generator 0 (PG0) output latch register (Can be read by reading port (P7) set for PG output.)				Shift alternate register 0 Register for PG mode (4-bit write)			

Don't use RMW instructions.

Figure 3.10 (3) Pattern Generator 0 Register (PG0REG)

PG1REG (004DH)

	7	6	5	4	3	2	1	0
bit Symbol	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
Read/Write	W				R / W			
After reset	0	0	0	0	Undefined			
Function	Pattern generator 1 (PG1) output latch register (Can be read by reading port (P7) set for PG output.)				Shift alternate register 1 Register for PG mode (4-bit write)			

Don't use RMW instructions.

Figure 3.10 (4) Pattern Generator 1 Register (PG1REG)

T89CR (003AH)

	7	6	5	4	3	2	1	0
bit Symbol	–				PG1T	PG0T	DBAEN	DB8EN
Read/Write	R / W				R / W			
After reset	0				0	0	0	0
Function	Fixed to "0". *Always read as "0".				G1 shift trigge 0: 8-bit timer trigger 3 (timers 2, 3) 1: 16-bit timer trigger (timer 9)	PG0 shift trigger 0: 8-bit timer trigger 1 (timers 0, 1) 1: 16-bit timer trigger (timer 8)	Double-buffer 0: Disable 1: Enable	
							TREGA double-buffer	TREG8 double-buffer

Double-buffer control

0	Disable
1	Enable

DBAEN : TREGA double-buffer
DB8EN : TREG8 double-buffer

PG0 shift trigger select

0	8-bit timer trigger (timers 0, 1)
1	16-bit timer trigger (timer 8)

PG1 shift trigger select

0	8-bit timer trigger (timers 2, 3)
1	16-bit timer trigger (timer 9)

When writing to other bits, always write "0" to this bit.

Figure 3.10 (5) 16-Bit Timer Trigger Control Register (T89CR)

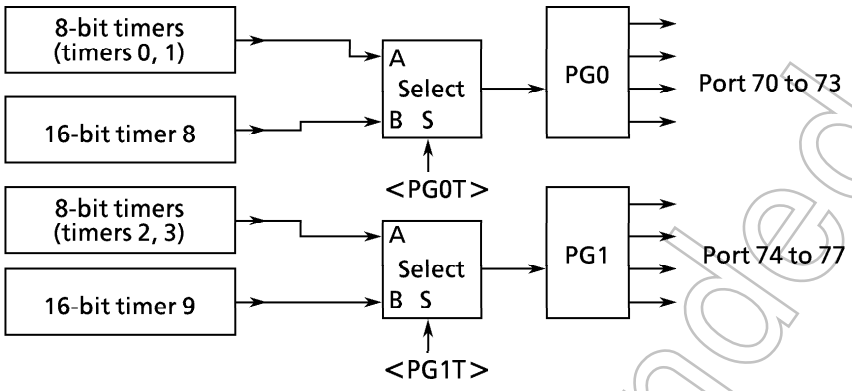


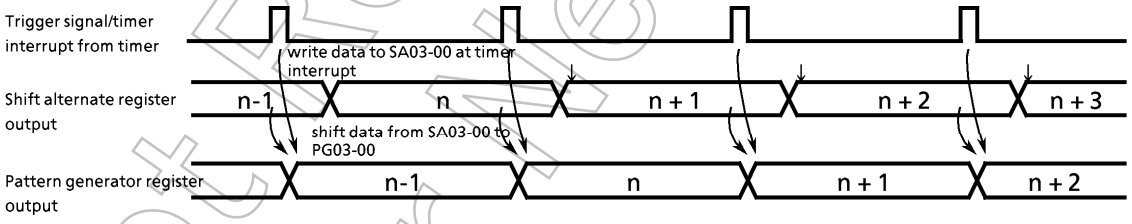
Figure 3.10 (6) Connections Between Timers and Pattern Generator

(1) Pattern Generation Mode

Setting the <PAT0> bit of PG01CR to "1" sets PG to pattern generation mode. In this mode, the CPU can only write to the shift alternate register. Therefore, by writing to the PG during processing of interrupts from the shift trigger timer, a pattern is output in real time in sync with the timer.

In pattern generation mode, set the <PG0M> bit of PG01CR to "1", the <CCW0> bit of PG01CR to "0", and the <PG0TE> bit of PG01CR to "1".

As the PG outputs to port 7, and the port 7 bits can be individually switched between port and function operation by the port 7 function register P7FC, any port pin can be set for PG output. Figure 3.10 (7) is a block diagram of this mode.



Example of Pattern Generation Mode Timing

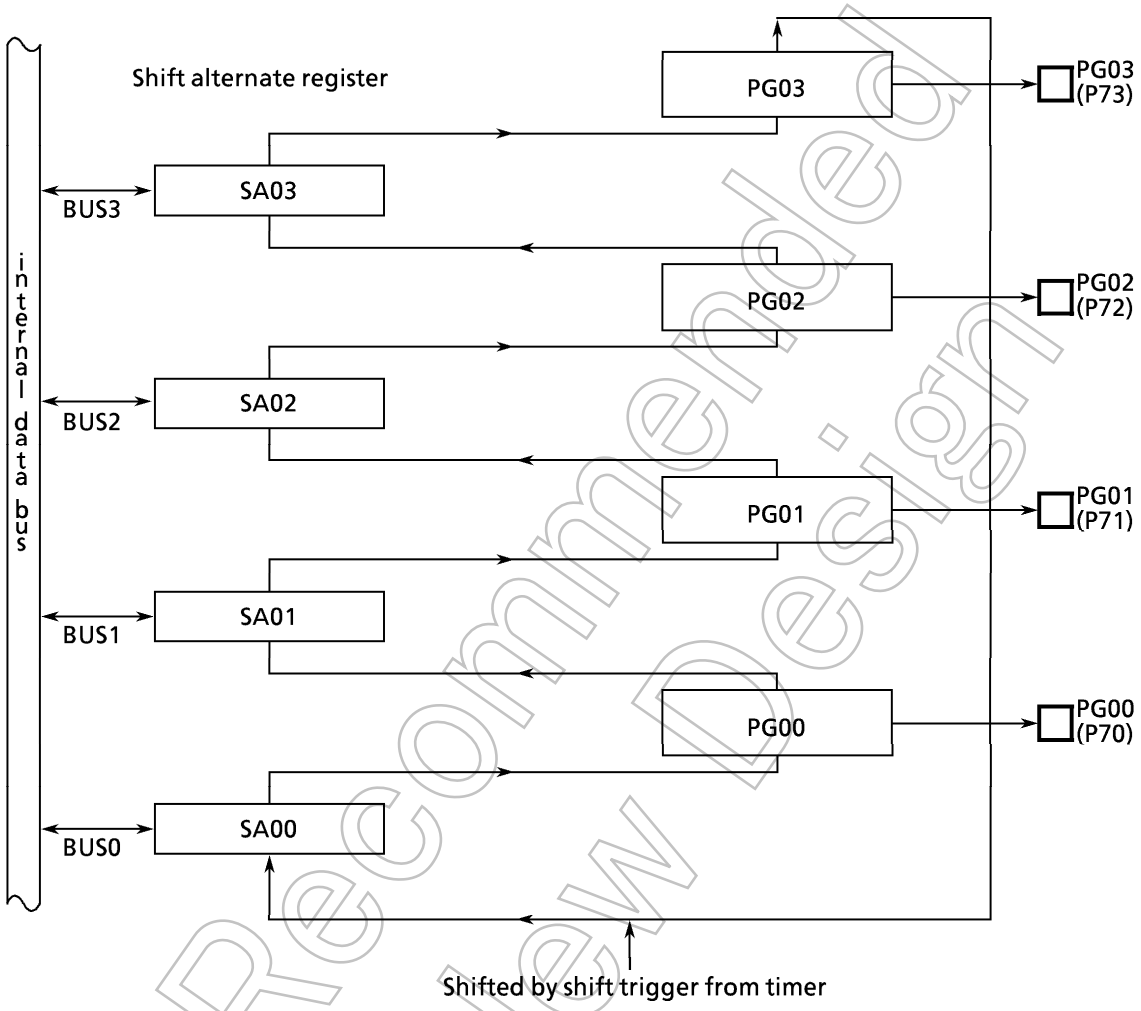


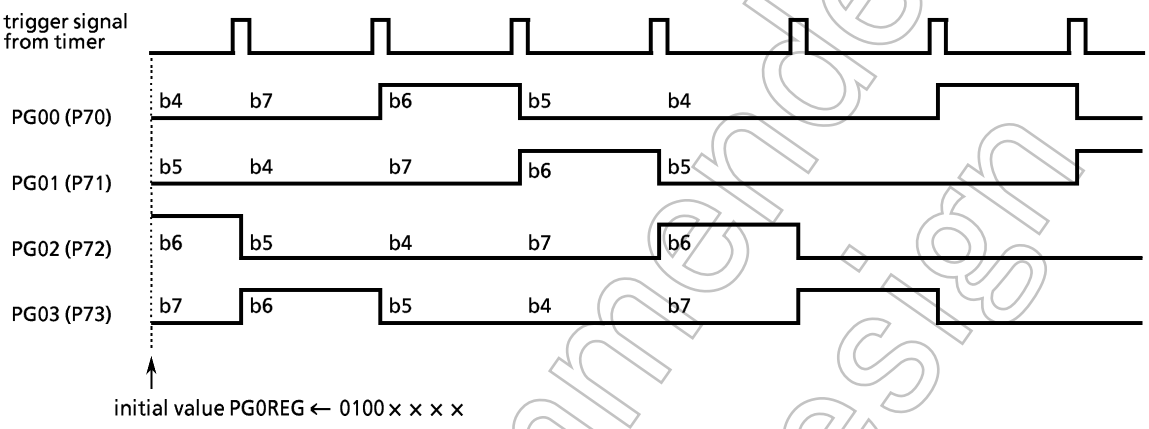
Figure 3,10 (7) Block Diagram of Pattern Generation Mode (PG0)

In pattern generation mode, writing to the output latch by hardware is disabled. Otherwise, operation is the same as 1-2 excitation in stepping motor control mode. Accordingly, when writing data after a shift due to a trigger signal from the timer, the data must be written before the next trigger signal.

(2) Stepping Motor Control Mode

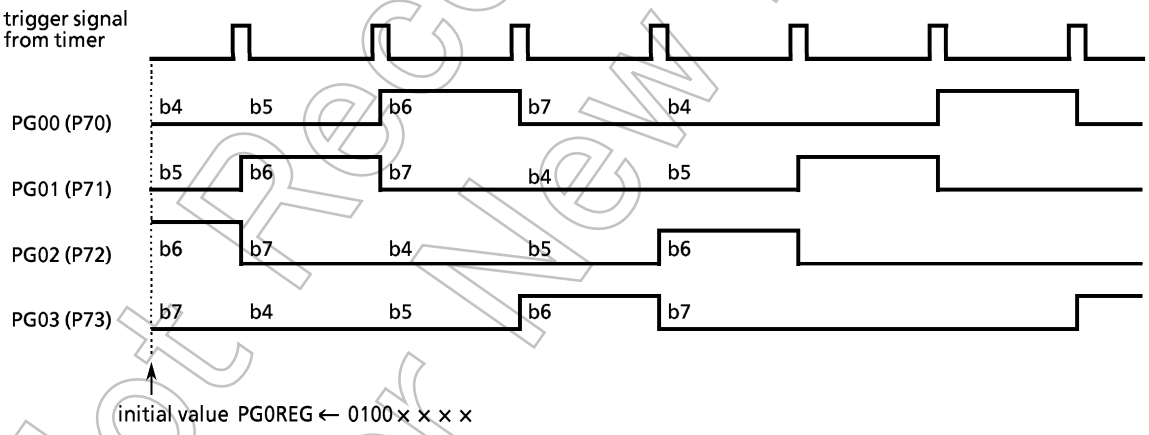
① 4-phase, 1 or 2 excitation

Figures 3.10 (8) and (9) show the channel 0 (PG0) output waveforms for 4-phase, 1 excitation, and 4-phase, 2 excitation.



Note: bn is the initial value for PG0REG <-- b7 b6 b5 b4 x x x x.

1. Normal



2. Reverse

010289

Note: bn is the initial value for PG0REG <-- b7 b6 b5 b4 x x x x.

Figure 3.10 (8) 4-Phase, 1 Excitation Output Waveform (Normal/Reverse)

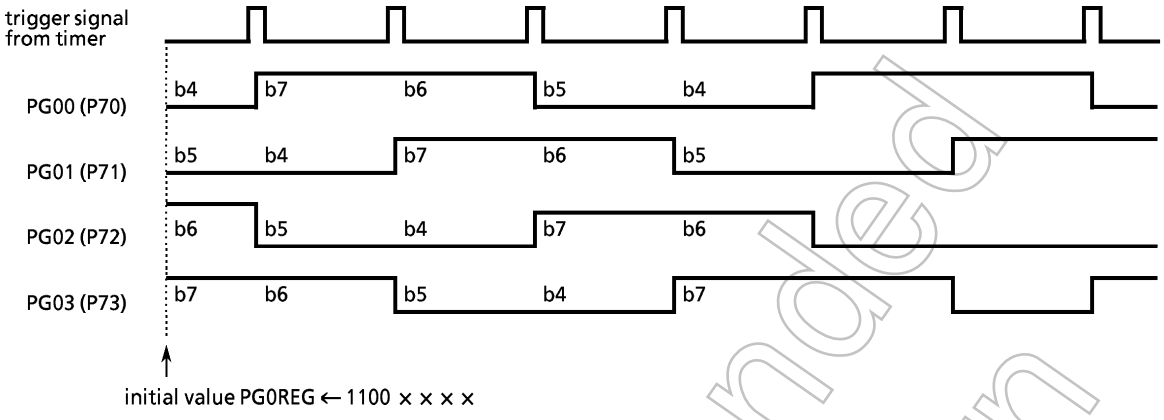


Figure 3.10 (9) 4-Phase, 2 Excitation Output Waveform (Normal)

The output latch of PG0 (also used as P7) is shifted on the rising edge of the trigger signal from the timer and output to the port.

The PG01CR<CCW0> bit sets the shift direction. Setting CCW0 to "0" sets the normal direction (PG00 --> PG01 --> PG02 --> PG03); setting to "1" sets the reverse direction (PG00 <-- PG01 <-- PG02 <-- PG03).

Setting one bit only to "1" when initializing PG results in 4-phase, 1 excitation. Setting two consecutive bits to "1" results in 4-phase, 2 excitation. When a 4-phase, 1 or 2 excitation waveform is output, the shift alternate register is ignored.

Figure 3.10 (10) is a block diagram of 4-phase, 1 or 2 excitation (normal).

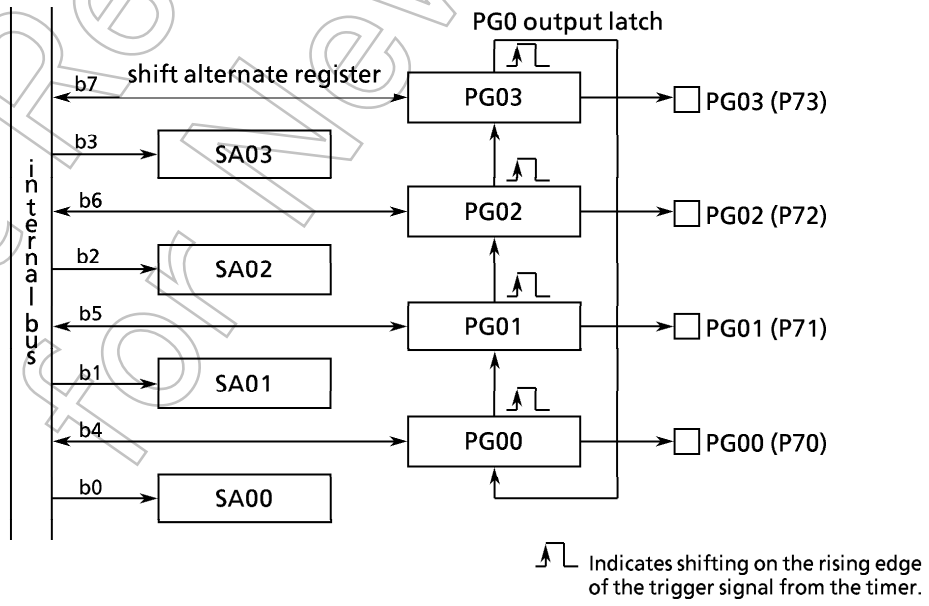
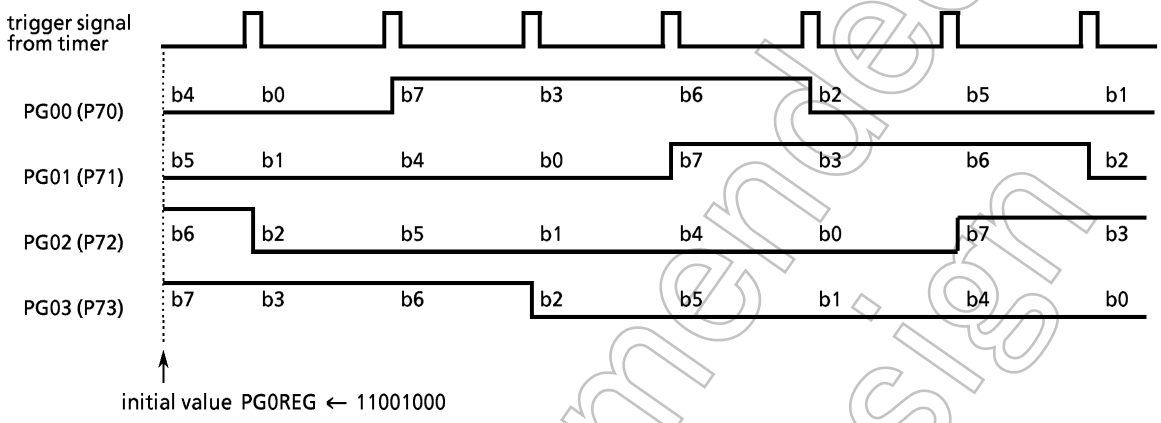


Figure 3.10 (10) Block Diagram of 4-Phase, 1 or 2 Excitation (Normal)

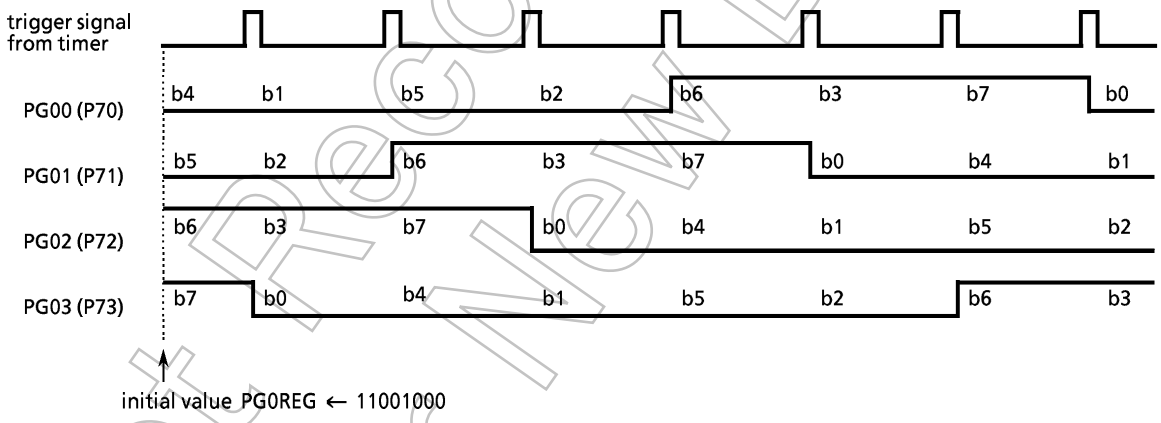
② 4-phase, 1-2 excitation

Figure 3.10 (11) shows the channel 0 (PG0) output waveforms for 4-phase, 1-2 excitation.



Note : bn is the initial value for PG0REG ← b7 b6 b5 b4 b3 b2 b1 b0.

① Normal



Note : bn is the initial value for PG0REG ← b7 b6 b5 b4 x x x x.

② Reverse

Figure 3.10 (11) 4-Phase, 1-2 Excitation Output Waveform (Normal/ Reverse)

The initial values to be set for 4-phase 1-2 excitation are as follows:

When the initial values | b7 b6 b5 b4 b3 b2 b1 b0 | are arranged as | b7 b3 b6 b2 b5 b1 b4 b0 |

set three consecutive bits to 1 and the rest to 0 (positive logic). For example, setting b7, b3, and b6 to 1 results in "11001000B" and the output waveform shown in Figure 3.10 (11) is obtained.

To output a negative logic waveform, invert the "1"s and "0"s of the initial value. For example, to reverse the logic of the waveform output in Figure 3.10 (11), set the initial value to "00110111".

The PG0 output latch (also used as P7) and the shift alternate register for the pattern generator (SA0) are shifted and output to the port on the rising edge of the trigger signal from the timer. PG01CR<CCW0> sets the shift direction.

Figure 3.10 (12) is a block diagram of 4-phase, 1-2 excitation (normal).

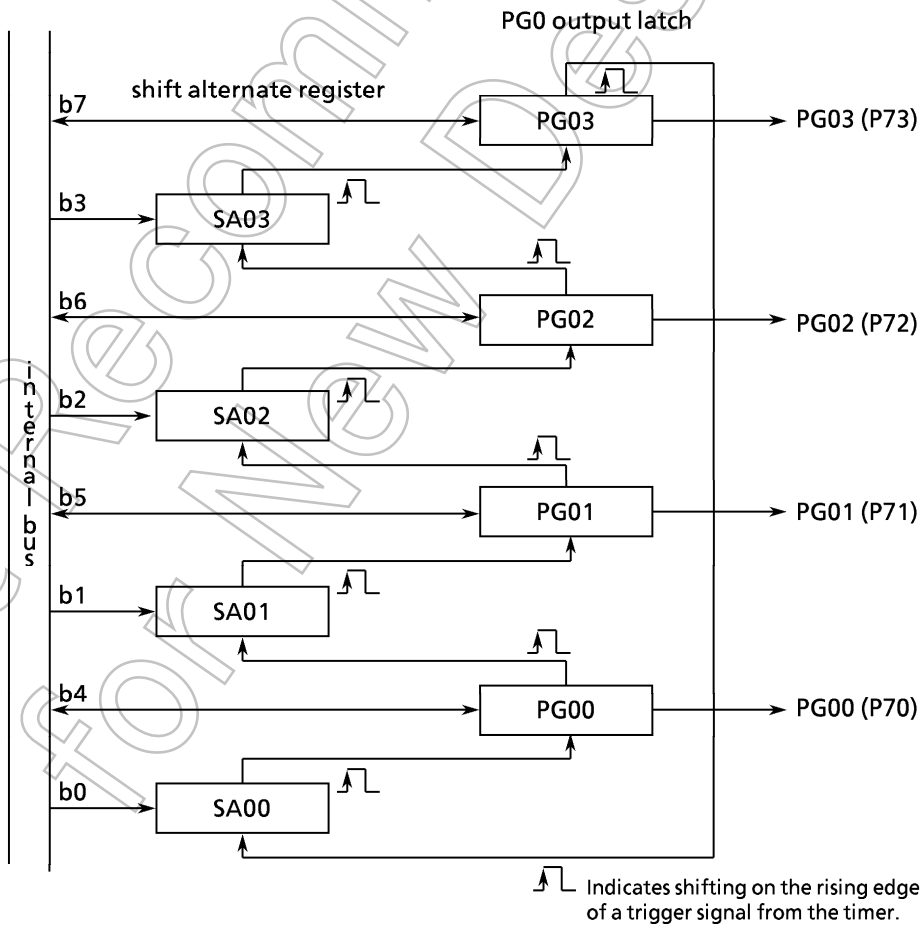


Figure 3.10 (12) 4-Phase, 1-2 Excitation (Normal)

Example : To drive a stepping motor from channel 0 (PG0) with 4-phase, 1-2 excitation (normal) using timer 0, set the registers as follows.

```
          7 6 5 4 3 2 1 0
T8RUN   ← - X - - - - - 0    Stops timer 0 and clears to zero.
T01MOD  ← 0 0 X X - - 0 1    Sets to 8-bit timer mode, inputs clock φT1.
T02FFCR ← X X X 0 1 0 1 0    Clears TFF1 and enables inversion trigger from timer 0.
TREG0   ← * * * * * * * *    Sets cycle in timer register.
P7CR    ← - - - - 1 1 1 1    Sets P70-P73 to output mode.
P7FC    ← - - - - 1 1 1 1    Sets P70-P73 to PG outputs.
PG01CR  ← - - - - 0 0 1 1    Sets PG0 to 4-phase, 1-2 excitation, normal direction.
PG0REG  ← 1 1 0 0 1 0 0 0    Sets the initial value.
T8RUN   ← - - - - - - - 1    Starts timer 0.
T16RUN  ← 1 X - - X X X X
```

Note : X ; Don't care - ; No change

(3) Trigger Signal from Timer

The trigger signals from the timers used by the PG differ from the trigger signals for timer flip-flop (TFF1, 3, 8, 9, A, B) inversion. Figure 3.10 (1) shows the differences in trigger signal timing for all 8-bit timer operating modes.

Table 3.10 (1) Trigger Signal Selection

	Invert TFF1	Shift PG
8-bit timer mode	Selected by T02FFCR<FF1/S> at match between up-counter and TREG0 or TREG1	Timing same as at left
16-bit timer mode	At match between up-counter and TREG0/TREG1 (up-counter value = TREG1* 2^8 + TREG0)	Timing same as at left
PPG output mode	At match between up-counter and TREG0 or up-counter and TREG1	At match between up-counter and TREG1 (PPG cycle)
PWM output mode	At match between up-counter and TREG0; PWM cycle	No trigger signal for PG shift generated

Note : To shift PG, set T02FFCR<FF1IE> = "1" to enable TFF1 inversion.

PG can synchronize with 16-bit timers T8 and T9. However, the 16-bit timer only outputs a PG shift trigger signal when the UC8/9 value matches TREG9/B value. When using a trigger signal from T8, set either T8FFCR<EQ9T8> or T8MOD<EQ9T9> to "1" to generate a trigger at a match with TREG9. When using a trigger signal from TA, set T9FFCR<EQBTA> to "1" to generate a trigger at a match with TREGB.

(4) PG and Timer Output Application

As described in (3), Trigger Signal from Timer, the PG shift and TFF invert timing depend on the timer mode. The following is an application example for operating the PG with an 8-bit timer in PPG output mode.

When driving a stepping motor, a sync signal is often required at the excitation switching to align each phase value (PG output). For this purpose, this application uses port 7 as the stepping motor control port and outputs a sync clock to TO1 (also used as P91).

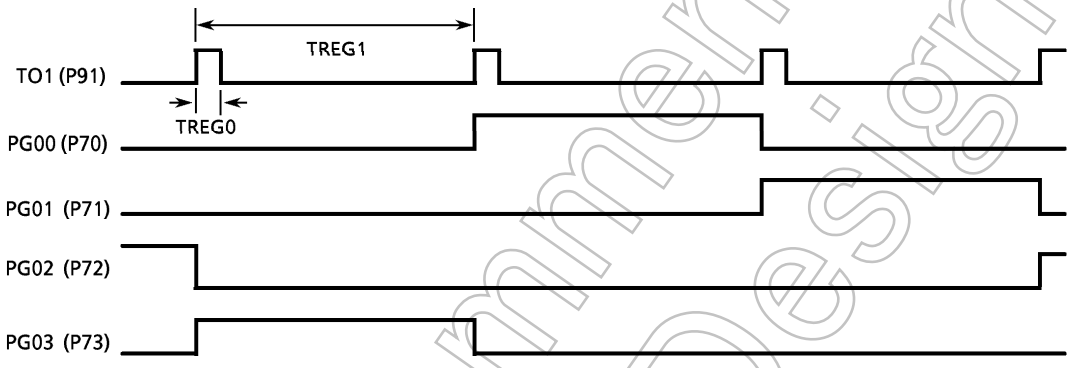


Figure 3.10 (13) 4-Phase, 1 Excitation Output Waveform

Setting example:

```
          7 6 5 4 3 2 1 0
T8RUN     ← - - - - - - 0 0   Stops timers 0 and 1, and clears to zero.
T01MOD    ← 1 0 X X X X 0 1   Sets timers 0 and 1 to PPG mode, inputs clock φT1.
T02FFCR   ← X X X 0 0 1 1 X   Sets to "1" to enable TFF1 inversion.
TREG0     ← * * * * * * * *   Sets TO1 duty.
TREG1     ← * * * * * * * *   Sets TO1 cycle.
P9CR      ← - - - - - - 1 -
P9FC      ← - - - - - - 1 X   } Sets P91 as TO1 pin.
P7CR      ← - - - - 1 1 1 1
P7FC      ← - - - - 1 1 1 1   } Sets P70 to P73 as PG0 pins.
PG01CR    ← - - - - 0 0 0 1   Sets PG0 to 4-phase, 1 excitation.
PG0REG    ← * * * * * * * *   Sets initial value.
T8RUN     ← - - - - - - 1 1   Starts timers 0 and 1.
T16RUN    ← 1 X - - X X X X
```

Note : X ; Don't care - ; No change

3.11 Serial Channel

TMP95C063 features two built-in serial input/output channels.
The serial channel operating modes are as follows:

- I/O interface mode — Mode 0 : For receiving and transmitting I/O data for I/O extension, and for receiving and transmitting synchronous I/O data signals (SCLK).
- Universal asynchronous receiver transmitter (UART) mode
  - Mode 1 : 7-bit transmit/receive data
  - Mode 2 : 8-bit transmit/receive data
  - Mode 3 : 9-bit transmit/receive data

Parity bits can be added in modes 1 and 2. Mode 3 has a wake-up function to start slave controllers using serially linked master controllers (multi-controller system).

Figure 3.11 (1) shows the data formats (for one frame) in each mode.

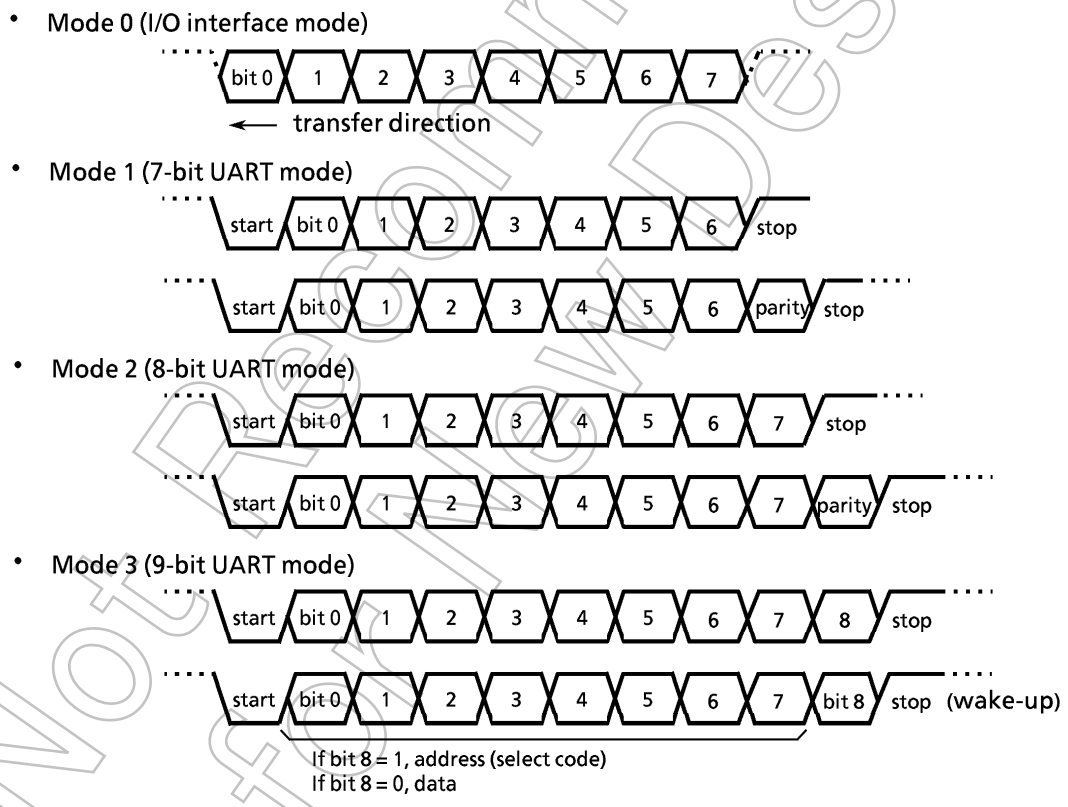


Figure 3.11 (1) Data Formats

Serial channel buffer registers temporarily hold data to be transmitted or received (full-duplex), allowing independent transmission and reception.

Note that in I/O interface mode, the serial clock (SCLK) is shared between reception and transmission (half-duplex).

The buffer register for reception features a double-buffer configuration to prevent overrun error; an extra frame holds data until the data are read by the CPU. That is, a receive buffer holds the data already received, while the buffer register receives the next frame of data.

By using $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ (as no $\overline{\text{RTS}}$ pin is provided, a pin in any port must be controlled by software), it is possible to halt data transmission until the CPU reads the data received after each frame (handshake function).

In UART mode, a check function prevents data receive operations from starting due to erroneous start bits being generated by noise or other interference on the line. The channel starts receiving data only when the start bit is detected as normal in at least two of three samplings.

When the transmit buffer is empty, an INTTX interrupt is generated to request the CPU to supply the next data to transmit. When the receive buffer has data to be read by the CPU, an INTRX interrupt is generated.

When an overrun error, parity error, or framing error is detected at data reception, the corresponding flag <OERR, PERR, FERR> is set in the control register (SC0CR/SC1CR) of the relevant serial channel.

Serial channels 0 and 1 have a dedicated baud rate generator, which can set any baud rate by dividing the frequency of internal input clocks (ϕT0, ϕT2, ϕT8, and ϕT32) from the 9-bit prescaler (shared with 8/16 bit timers) by a value between 1 and 16.

In addition to the clock from the internal baud rate generator, an arbitrary baud rate can be obtained from the external clock input (SCLK). Moreover, in I/O interface mode, a sync signal (SCLK) can be input and data transfer performed using this external clock.

3.11.1 Control Registers

Each serial channel is controlled by three control registers (SC0CR, SC0MOD, and BR0CR for channel 0). Transmit/receive data are stored in a register in each channel (SC0BUF for channel 0).

SC0MOD (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transmit data bit 8	Hand shake function control 0: CTS disable 1: CTS enable	Receive control 0: receive disable 1: receive enable	Wake-up function 0: disable 1: enable	Serial transfer mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transfer clock (UART) 00: TO2 trigger 01: baud rate generator 10: internal clock Φ1 11: external clock (SCLK0)	

Serial transfer clock (UART)

00	Timer 2 match detect signal
01	Baud rate generator
10	Internal clock Φ1
11	External clock (SCLK0)

Serial transfer mode

00	I/O interface mode	
01	UART mode	7-bit
10		8-bit
11		9-bit

Wake-up function

	9-bit UART	Other modes
0	Interrupt if receive	Don't care
1	Interrupt only when RB8 = 1	

Receive control

0	Receive disable
1	Receive enable

Handshake function ($\overline{CTS}$ pin) enable

0	Disable (transmission always enabled)
1	Enable

Transmit data bit 8

Note :SC1MOD (56H) is provided for channel 1.

Figure 3.11 (2) Serial Mode Control Register (SC0MOD, Channel 0)

SC0CR (0051H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to "0" when read.)			R/W	
After reset		0	0	0	0	0	0	0
Function	Receive data bit 8	Parity 0: Odd 1: Even	Parity addition 0: disable 1: enable	1 : Error Overrun	Parity	Framing	0: SCLK0 (↑) 1: SCLK0 (↓)	0: Baud rate generator 1: SCLK0 pin input

I/O interface input clock select

0	Baud rate generator
1	SCLK0 pin input

Edge select in SCLK0 input operations

0	Data transmit/receive (↑) at SCLK0 rising edge
1	Data transmit/receive (↓) at SCLK0 falling edge

Framing error flag
Parity error flag
Overrun error flag
Cleared to "0" when read.

Parity addition enable

0	Disable
1	Enable

Even parity addition/check

0	Odd parity
1	Even parity

Receive data bit 8

Note : SC1CR (55H) is provided for channel 1.
As the error flags are all cleared after reading, when testing with a bit test instruction, test more than just a single bit.

Figure 3.11 (3) Serial Control Register (SC0CR, Channel 0)

BR0CR (0053H)

	7	6	5	4	3	2	1	0
bit Symbol			BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write			R/W					
After reset	0		0	0	0	0	0	0
Function	Fixed to "0".		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Sets divisor.			

Sets baud rate generator divisor.

0000	Divide by 16
0001	Divide by 1 (no division)
0010 to 1111	Divide by 2 to 15

Baud rate generator input clock select

00	Internal clock ϕT0 (4/fc)
01	Internal clock ϕT2 (16/fc)
10	Internal clock ϕT8 (64/fc)
11	Internal clock ϕT32 (256/fc)

Note : BR1CR (57H) is provided for channel 1.
To use the baud rate generator, set T16RUN<PRRUN> to "1" and run the prescaler.
The baud rate generator frequency can be divided by 1 in UART mode only. Do not use this setting in I/O interface mode.
Don't read from or write to BR0CR register during sending or receiving.

Figure 3.11 (4) Baud Rate Generator Control Register (BR0CR, Channel 0)

SC0BUF (50H)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(transmit)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(receive)

Figure 3.11 (5) Serial Transmit/Receive Register (SC0BUF, Channel 0)

SC1MOD (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transmit data bit 8	Handshake function control 0: CTS disable 1: CTS enable	Receive control 0: receive disable 1: receive enable	Wake-up function 0: disable 1: enable	Serial transfer mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transfer clock (UART) 00: TO2 trigger 01: baud rate generator 10: internal clock Φ1 11: external clock SCLK1	

Serial transfer clock (UART)

00	Timer 2 match detect signal
01	Baud rate generator
10	Internal clock Φ1
11	External clock SCLK1

Serial transfer mode

00	I/O interface mode	
01	UART mode	7-bit
10		8-bit
11		9-bit

Wake-up function

	9-bit UART	Other modes
0	Interrupt if receive	Don't care
1	Interrupt only when RB8 = 1	

Receive control

0	Receive disable
1	Receive enable

Handshake function ($\overline{CTS}$ pin) enable

0	Disable (transmission always enabled)
1	Enable

Transmit data bit 8

Figure 3.11 (6) Serial Mode Control Register (SC1MOD, Channel 1)

SC1CR (0055H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to "0" when read.)			R/W	
After reset		0	0	0	0	0	0	0
Function	Receive data bit 8	Parity 0: Odd 1: Even	Parity addition 0: disable 1: enable	1: Error			0: SCLK1 (↑) 1: SCLK1 (↓)	0: Baud rate generator 1: SCLK1 pin input
				Overrun	Parity	Framing		

I/O interface input clock select

0	Baud rate generator
1	SCLK1 pin input

Edge select in SCLK1 input operations

0	Data transmit/receive (↑) at SCLK1 rising edge
1	Data transmit/receive (↓) at SCLK1 falling edge

Framing error flag
Parity error flag
Overrun error flag
} Cleared to "0" when read

Parity addition enable

0	Disable
1	Enable

Even parity addition/check

0	Odd parity
1	Even parity

Receive data bit 8

Note : As the error flags are all cleared after reading, when testing with a bit test instruction, test more than just a single bit.

Figure 3.11 (7) Serial Control Register (SC1CR, Channel 1)

BR1CR (0057H)

	7	6	5	4	3	2	1	0
bit Symbol			BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write			R/W					
After reset	0		0	0	0	0	0	0
Function	Fixed to "0".		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Sets divisor.			

Baud rate generator divisor setting

0000	Divide by 16
0001	Divide by 1 (no division)
0010 to 1111	Divide by 2 to 15

Baud rate generator input clock selection

00	Internal clock ϕT0 (4/fc)
01	Internal clock ϕT2 (16/fc)
10	Internal clock ϕT8 (64/fc)
11	Internal clock ϕT32 (256/fc)

Note : To use the baud rate generator, set T16RUN<PRRUN> to "1" and run the prescaler. The baud rate generator frequency can be divided by 1 in UART mode only. Do not use this setting in I/O interface mode.
Don't read from or write to BR1CR register during sending or receiving.

Figure 3.11 (8) Baud Rate Generator Control Register (BR1CR, Channel 1)

SC1BUF (0054H)

	7	6	5	4	3	2	1	0	
	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(transmit)
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(receive)

Figure 3.11 (9) Serial Transmit/Receive Buffer Register (SC1BUF, Channel 1)

PAFC (0081H)

	7	6	5	4	3	2	1	0
bit Symbol	PA7F			PA4F	PA3F			PA0F
Read/Write	W			W	W			W
After reset	0			0	0			0
Function	0: PORT 1: SCLK1			0: PORT 1: TxD1	0: PORT 1: SCLK0			0: PORT 1: TxD0

Don't use RMW instructions.

Sets PA0 to TxD0 output.

0	Port output
1	TxD0 (channel 0) output

Sets PA3 to SCLK output.

0	Port output
1	SCLK0 (channel 0) output

Sets PA4 to TxD1 output.

0	Port output
1	TxD1 (channel 1) output

Sets PA7 to SCLK1 output.

0	Port output
1	SCLK1 (channel 1) output

Figure 3.11 (10) Port A Function Register (PAFC)

ODE (0058H)

	7	6	5	4	3	2	1	0
bit Symbol							ODE1	ODE0
Read/Write							R/W	
After reset							0	0
Function							PA4 0: CMOS 1: OPEN Drain	PA0 0: CMOS 1: OPEN Drain

Sets PA0 to open drain output.

0	CMOS output
1	Open drain output

Sets PA4 to open drain output.

0	CMOS output
1	Open drain output

Figure 3.11 (11) Port A Open Drain Enable Register (ODE)

3.11.2 Configuration

Figure 3.11 (12) is a block diagram of serial channel 0. Serial channel 1 has the same circuit configuration.

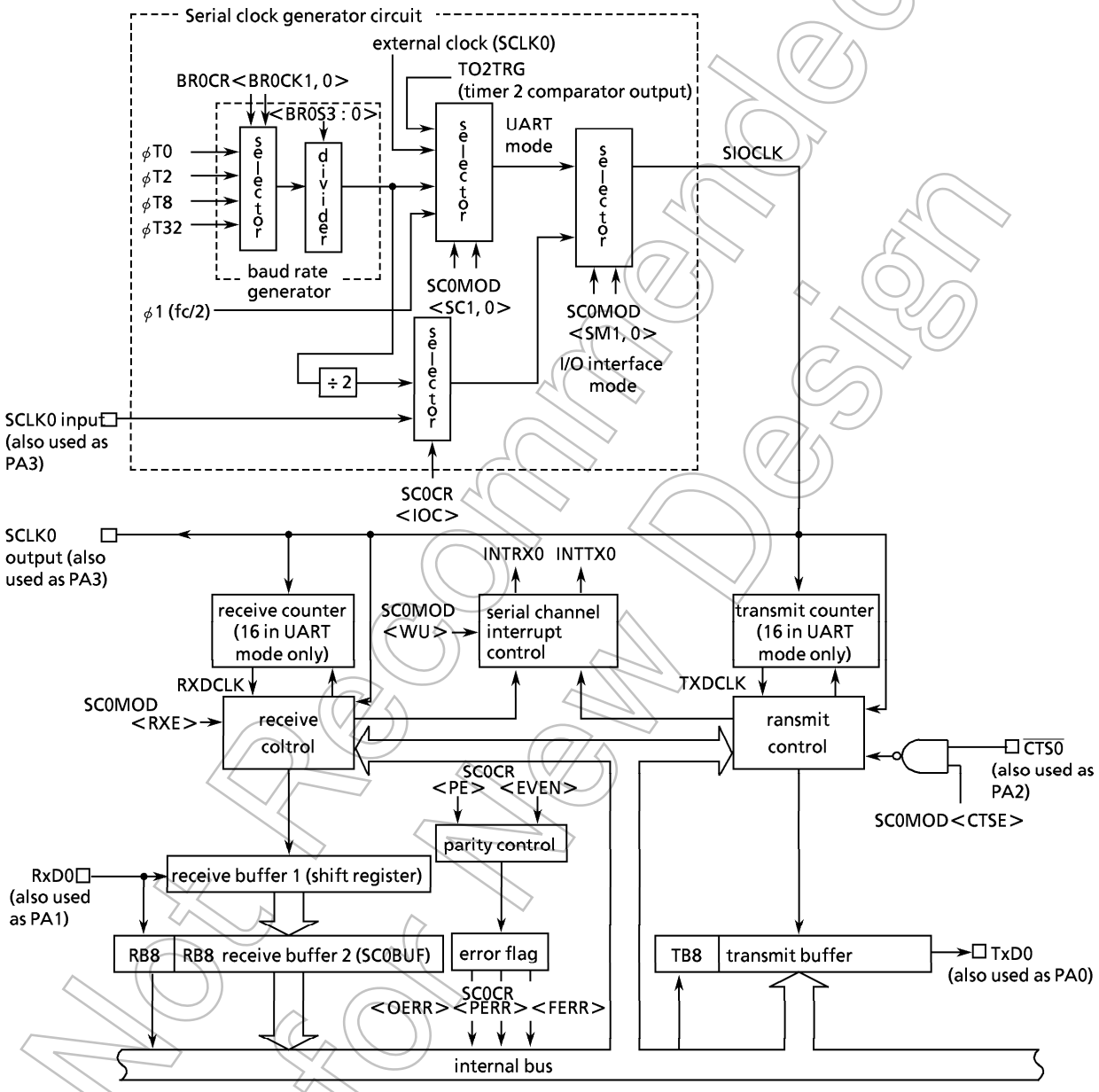


Figure 3.11 (12) Serial Channel 0 Block Diagram

① Baud rate generator

The baud rate generator is a circuit to generate the transmission clock signals that control the serial channel transmission rate.

The baud rate generator input clock is one of ϕT0 (4/fc), ϕT2 (16/fc), ϕT8 (64/fc), or ϕT32 (256/fc) from the 9-bit prescaler that the baud rate generator shares with the timers.

Bits 5 and 4 <BR0CK1, 0>/<BR1CK1, 0> of the baud rate generator control register (BR0CR/BR1CR) select the input clock.

The baud rate generator features a built-in 4-bit divider. Set the transmission rate by dividing the frequency by 1 to 16 using the divider.

Baud rates using the baud rate generator are determined as follows:

- UART mode

$$\text{Baud Rate} = \frac{\text{baud rate generator input clock}}{\text{baud rate generator divisor}} \div 16$$

- I/O interface mode

$$\text{Baud Rate} = \frac{\text{baud rate generator input clock}}{\text{baud rate generator divisor}} \div 2$$

The relationship between the input clock and the source clock (fc) is:

ϕT0 = 4/fc
ϕT2 = 16/fc
ϕT8 = 64/fc
ϕT32 = 256/fc

Accordingly,with the source clock set to 12.288 MHz, when ϕT2 (16/fc) is selected as input clock and the divisor is 5, the baud rate in UART mode is:

$$\text{Baud Rate} = \frac{fc/16}{5} \div 16$$
$$= 12.288 \times 10^6 \div 16 \div 5 \div 16 = 9600 \text{ (bps)}$$

Table 3.11 (1) shows examples of the baud rates in UART mode.

In UART mode, the serial channels use 8-bit timer 2 to obtain the baud rate. Table 3.11 (2) shows examples of baud rates using timer 2.

Moreover, the external clock input can also be used as the serial clock. The baud rate in this case is determined as follows.

Baud rate = external clock input ÷ 16

Table 3.11 (1) UART Mode Baud Rate Selection (1) (Using Baud Rate Generator)

Unit: Kbps

fc [MHz]	Input Clock / Divisor	ϕT0 (4/fc)	ϕT2 (16/fc)	ϕT8 (64/fc)	ϕT32 (256/fc)
9.830400	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	16	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
↑	A	19.200	4.800	1.200	0.300
14.745600	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	C	19.200	4.800	1.200	0.300

Note : In I/O interface mode, the transmission rate is eight times the values shown in this table.

Table 3.11 (2) UART Mode Baud Rate Selection (2) (Using Timer 2 Input Clock ϕT1)

Unit: Kbps

TREG2 \ fc	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	96		76.8	62.5	48
2H	48		38.4	31.25	24
3H	32	31.25			16
4H	24		19.2		12
5H	19.2				9.6
8H	12		9.6		6
AH	9.6				4.8
10H	6		4.8		3
14H	4.8				2.4

Baud rate calculation (using timer 2):

$$\text{Transmission rate:} = \frac{fc}{TREG2 \times 8 \times 16}$$

(Where timer 2 input clock is ϕT1)

Input clocks for timer 0

ϕT1 = 8/fc

ϕT4 = 32/fc

ϕT16 = 128/fc

Note : In I/O interface mode, the timer 2 match signal cannot be used as a transmission clock.

② Serial clock generator circuit

This circuit generates the transmit/receive basic clock.

- In I/O Interface mode

  In SCLK output mode where SC0CR/SC1CR<IOC> is set to "0", the basic clock (SIOCLK) is generated by dividing the output of the baud rate generator by 2.

  In SCLK input mode where SC0CR/SC1CR<IOC> is set to "1", the basic clock is derived from the rising or falling edge of the SCLK input, as determined by the setting of the SC0CR/SC1CR<SCLKS> register.

- In universal asynchronous receiver transmitter(UART) mode

  Basic clock SIOCLK is selected from one of the following depending on the setting of the <SC1,0> bits of the SC0MOD or SC1MOD register: the clock from the baud rate generator, internal clock $\phi 1$ (500K bps @ fc=16M Hz), a match detect signal from timer 2, or an external clock.

③ Receive counter

The receive counter is a 4-bit binary counter that counts by the SIOCLK clock and is used in universal asynchronous receiver transmitter (UART) mode. Sixteen cycles of SIOCLK are used to receive one bit of data. The data are sampled three times: at the 7th, 8th, and 9th clock cycles.

The data received are checked by the majority rule applied to the three samples.

For example, if the sampled data bits are 1, 0, 1 at the 7th, 8th, and 9th clock cycles respectively, the data are determined as "1". If the samplings are 0, 0, 1, the data received are determined as "0".

④ Receive control section

- In I/O Interface mode

  In SCLK output mode where SC0CR/SC1CR<IOC> is set to "0", the RxD0/1 pin is sampled at the rising edge of the shift clock output on the SCLK0/1 pin.

  In SCLK input mode where SC0CR/SC1CR<IOC> is set to "1", the RxD0/1 pin is sampled at the rising or falling edge of SCLK input as determined by the setting of the SC0CR/SC1CR<SCLKS> register.

- In universal asynchronous receiver transmitter (UART) mode

  The receive control section has a circuit for detecting the start bit by the majority rule. If two or more 0s are detected among three samples, the circuit recognizes the bit as a start bit and begins receiving. Data being received are also checked by the majority rule.

⑤ Receive buffer

The receive buffer has a double-buffer configuration to prevent overrun error. Receive buffer 1 (a shift register buffer) stores the data received bit by bit. When the receive buffer contains seven or eight bits of data, the data are transferred to receive buffer 2 (SC0BUF/SC1BUF), generating interrupt INTRX0/INTRX1.

The CPU reads only receive buffer 2 (SC0BUF/SC1BUF). Data can be stored in receive buffer 1 even before the CPU reads receive buffer 2.

However, receive buffer 2 must be read before all bits of the next data unit are received by buffer 1. Otherwise, an overrun error occurs and the contents of receive buffer 1 are lost, although the contents of receive buffer 2 and SC0CR<RB8>/SC1CR<RB8> are preserved. Reading receive buffer 2 (SC0BUF/SC1BUF) clears interrupt request flags INTRX0<IRX0C> and INTRX1<IRX1C>.

In 8-bit UART mode with parity added, the parity bit is stored in SC0CR<RB8>/SC1CR<RB8>. In 9-bit UART mode, the MSB is stored in SC0CR<RB8>/SC1CR<RB8>.

Setting SC0MOD<WU>/SC1MOD<WU> to "1" in 9-bit UART mode enables the slave controller wake-up. Only when SC0CR<RB8>/SC1CR<RB8> is set to 1, interrupt INTRX0/INTRX1 is generated.

⑥ Transmit counter

The transmit counter is a 4-bit binary counter for use in universal asynchronous receiver transmitter (UART) mode. Like the receive counter, the transmit counter counts by the SIOCLK clock, generating transmission clock TxDCLK every 16 clock cycles.

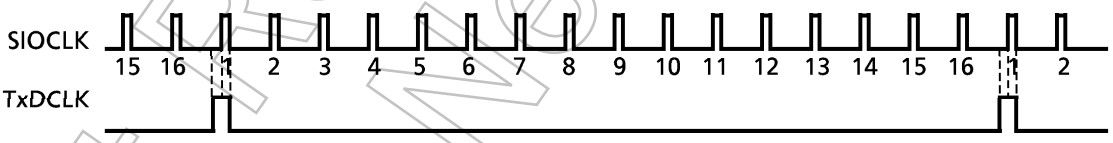


Figure 3.11 (13) Transmission Clock Generation

⑦ Transmit control section

- In I/O interface mode

In SCLK output mode where SC0CR/SC1CR<IOC> is set to "0", the data in the transmit buffer is output bit by bit to the TxD0/1 pin at the rising edge of the shift clock output on the SCLK0/1 pin.

In SCLK input mode where SC0CR/SC1CR<IOC> is set to "1", the data in the transmit buffer is output bit by bit to the TxD0/1 pin at the rising or falling edge of SCLK input as determined by the setting of the SC0CR/SC1CR<SCLKS> register.

· In universal asynchronous receiver transmitter (UART) mode

When the CPU writes data in the transmit buffer, transmission begins from the next rising edge of the TxDCLK, generating transmission shift clock TxDSFT.

Handshake Function

The serial channels use the $\overline{CTS}$ pin to transmit data in units of frames, thus preventing an overrun error. Use SC0MOD/SC1MOD<CTSE> to enable or disable the handshake function.

When $\overline{CTS}$ goes high, data transmission is halted after the completion of the current transmission and is not restarted until $\overline{CTS}$ returns to low. An INTTX0 interrupt is generated to request the CPU for the next data to transmit. When the CPU writes the data to the transmit buffer, processing enters standby mode.

An $\overline{RTS}$ pin is not provided, but a handshake function can easily be configured if the receiver sets any port assigned to the $\overline{RTS}$ function to high (in the receive interrupt routine) after data receive, and requests the transmitter to temporarily halt transmission.

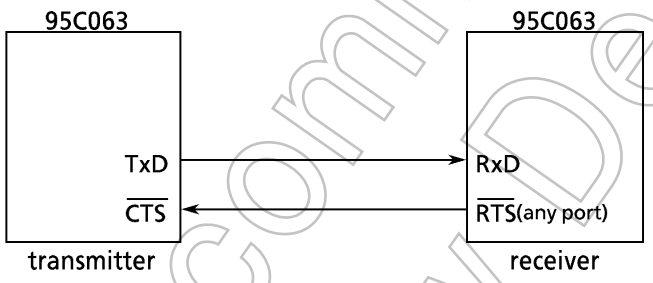
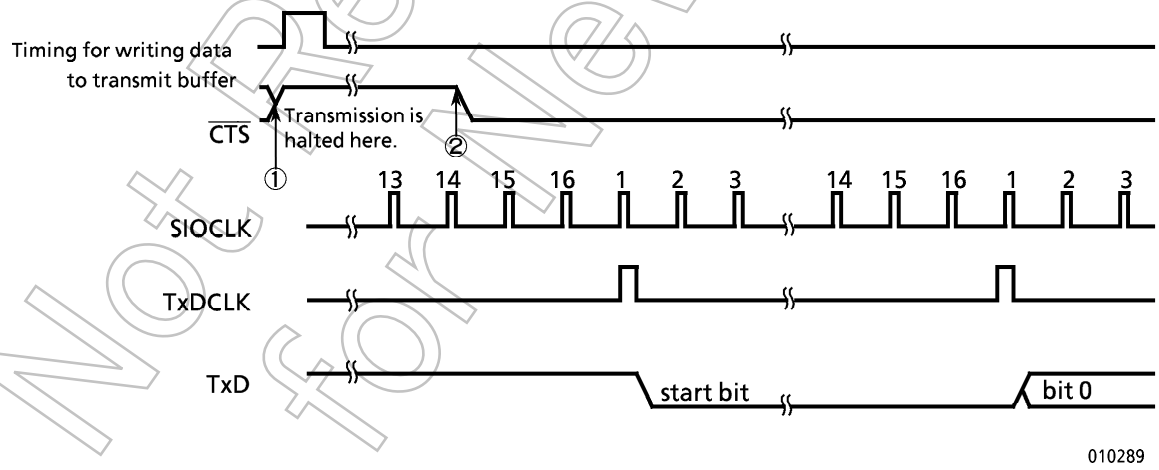


Figure 3.11 (14) Handshake Function



Note : ① When the $\overline{CTS}$ signal rises during transmission, transmission of the next data frame halts after transmission of the current data frame is complete.

② Transmission begins at the first TxDCLK clock falling edge after the $\overline{CTS}$ signal falls.

Figure 3.11 (15) $\overline{CTS}$ (Clear to Send) Signal Timing

⑧ Transmit buffer

Transmit buffer (SC0BUF/SC1BUF) shifts out and transmits the transmit data written by the CPU, beginning with the least significant bit, using the transmission shift clock (TxDSFT) generated by the transmission control section. When all bits are shifted out, the empty transmit buffer generates interrupt INTTX0/INTTX1.

⑨ Parity control circuit

When serial channel control register SC0CR<PE>/SC1CR<PE> is set to "1", data are transmitted and received with parity. However, parity can be added only in 7-bit or 8-bit UART mode. The SC0CR<EVEN>/SC1CR<EVEN> register selects even/odd parity.

At transmission, the parity control circuit automatically generates parity according to the data written in the transmit buffer (SC0BUF/SC1BUF). In 7-bit UART mode, the parity bit is stored in SC0BUF<TB7>/SC1BUF<TB7> prior to transmission. In 8-bit UART mode, parity is stored in SC0MOD<TB8>/ SC1MOD<TB8> prior to transmission. Set both <PE> and <EVEN> before writing the transmit data in the transmit buffer.

At receiving, data are first shifted into receive buffer 1. The parity control circuit automatically generates parity according to the data transferred to receive buffer 2 (SC0BUF/SC1BUF). In 7-bit UART mode, the generated parity is compared with the received parity in SC0BUF<RB7>/SC1BUF<RB7>. In 8-bit UART mode, the generated parity is compared with the received parity in SC0CR<RB8>/SC1CR<RB8>. If the parities differ, a parity error occurs and the SC0CR<PERR>/SC1CR<PERR> flag is set.

⑩ Error flags

Three error flags improve the reliability of data reception.

1. Overrun error <OERR>

When all bits of the next data frame have been received in receive buffer 1 while valid data are stored in receive buffer 2 (SCBUF0/1), an overrun error occurs.

2. Parity error <PERR>

The parity generated according to the data shifted into receive buffer 2 (SCBUF0/1) is compared with the parity bit received from the RxD pin. If the parities are not equal, a parity error occurs.

3. Framing error <FERR>

The stop bit of data received is sampled three times around the center. If the majority of the samples are "0", a framing error occurs.

⑪ Signal Generation Timing

1) In UART Mode

Receive

Mode	9 Bit	8 Bit + Parity	8 Bit, 7 Bit + Parity, 7 Bit
Interrupt generation timing	Center of last bit (bit 8)	Center of last bit (parity bit)	Center of stop bit
Framing error generation timing	Center of stop bit	Center of stop bit	Center of stop bit
Parity error generation timing	———	Center of last bit (parity bit)	←
Overrun error generation timing	Center of last bit (bit 8)	Center of last bit (parity bit)	Center of stop bit

Transmit

Mode	9 Bit	8 Bit + Parity	8 Bit, 7 Bit + Parity, 7 Bit
Interrupt generation timing	Immediately before stop bit is sent	←	←

270289

2) In I/O Interface Mode

Transmission interrupt generation timing	SCLK output mode	Immediately after rise of last SCLK signal (See Figure 3.11 (19))
	SCLK input mode	Immediately after rise of last SCLK signal (rising mode), immediately after fall in falling mode (See Figure 3.11 (20))
Receive interrupt generation timing	SCLK output mode	When received data are transferred to receive buffer 2 (SC0BUF/SC1BUF) (immediately after final SCLK) (See Figure 3.11 (21))
	SCLK input mode	When received data are transferred to receive buffer 2 (SC0BUF/SC1BUF) (immediately after final SCLK) (See Figure 3.11 (22))

3.11.3 Operation

(1) Mode 0 (I/O Interface Mode)

This mode is used to increase the number of I/O pins for transmitting or receiving data to an external shift register or other external destinations.

This mode consists of SCLK output mode for outputting a synchronous clock (SCLK), and SCLK input mode for inputting a synchronous clock (SCLK) from an external source.

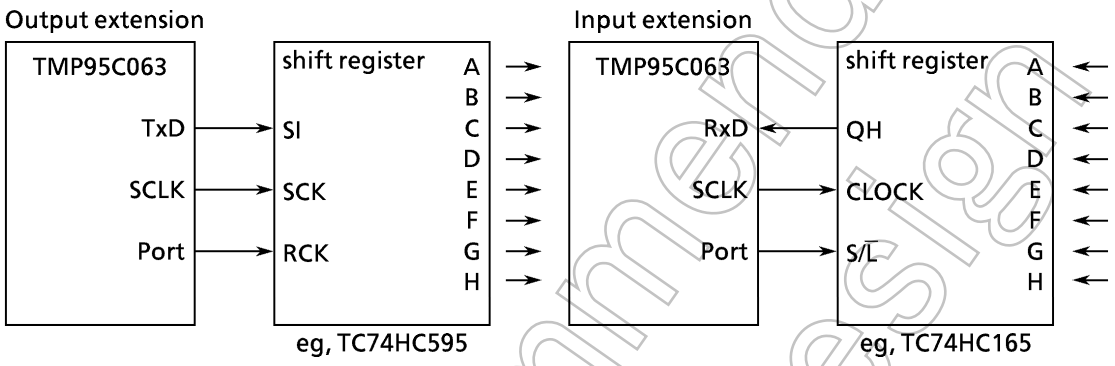


Figure 3.11 (16) Example of SCLK Output Mode Connection

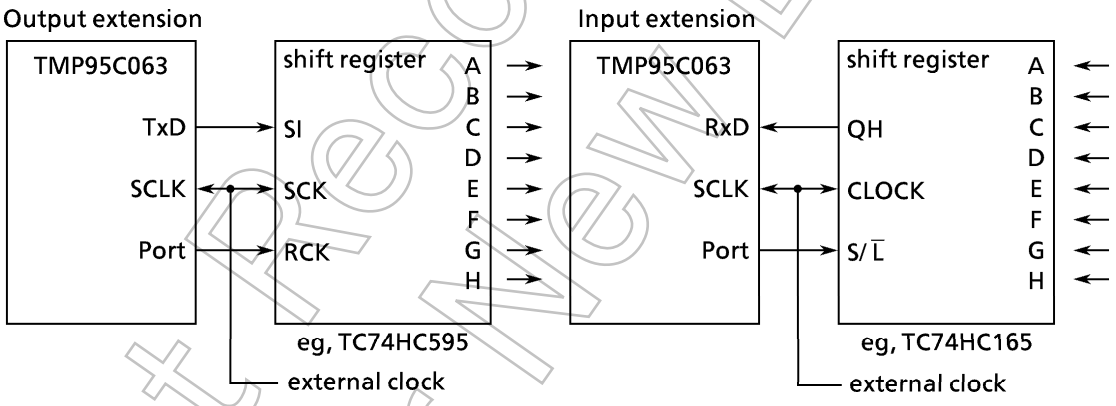


Figure 3.11 (17) Example of SCLK Input Mode Connection

① Transmission

In SCLK output mode, each time the CPU transmits data to the transmit buffer, eight data bits are output from the TxD0/1 pin, and a synchronous clock signal is output from the SCLK0/1 pin. When all data are output, INTES0<ITX0C> / INTES1<ITX1C> is set, generating interrupt INTTX0/1.

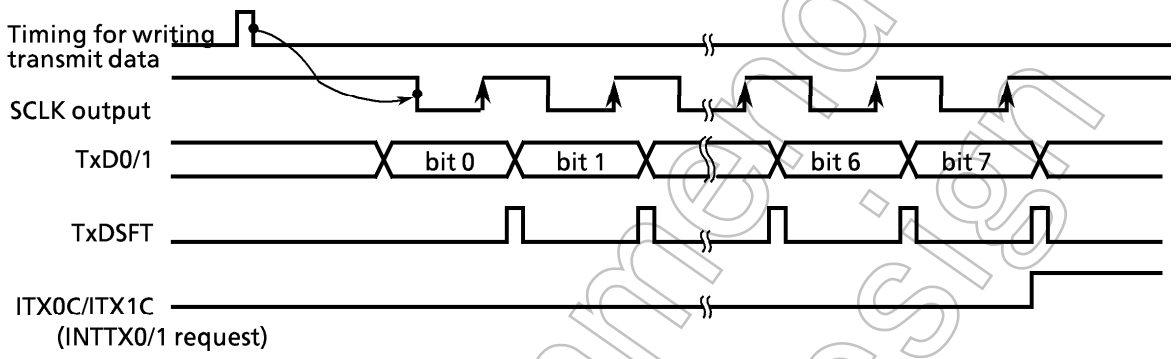


Figure 3.11 (18) Data Transmission in I/O Interface Mode (SCLK Output Mode)

In SCLK input mode, 8-bit data are output from TxD0/1 pin when SCLK input becomes active while data are written in the transmission buffer by CPU.

When all data are output, INTES0<ITX0C>/INTES1<ITX1C> is set, generating interrupt INTTX0/1.

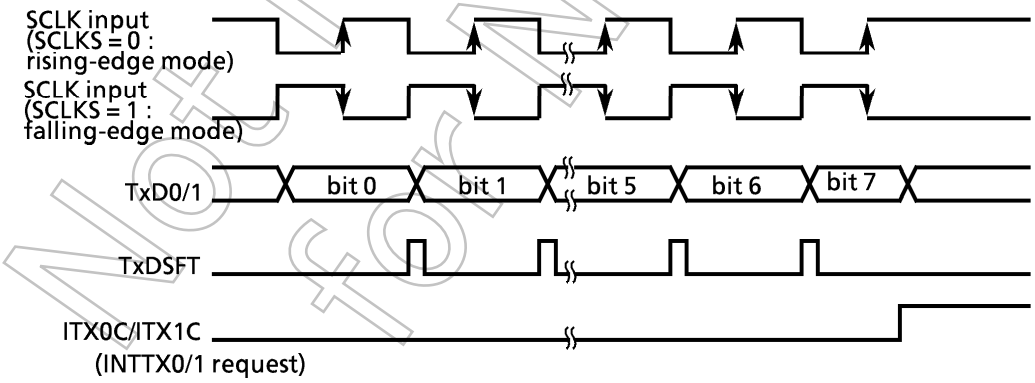


Figure 3.11 (19) Data Transmission in I/O Interface Mode (SCLK Input Mode)

② Receiving

In SCLK output mode, whenever the CPU reads the received data and clears the receive interrupt flag INTES0<IRX0C>/INTES1<IRX1C>, a synchronous clock is output from the SCLK0/1 pin and the next data frame is shifted to receive buffer 1. When an 8-bit data frame has been received, it is transferred to receive buffer 2 (SC0BUF/SC1BUF), and INTES0<IRX0C>/INTES1<IRX1C> is set again, generating interrupt INTRX0/1.

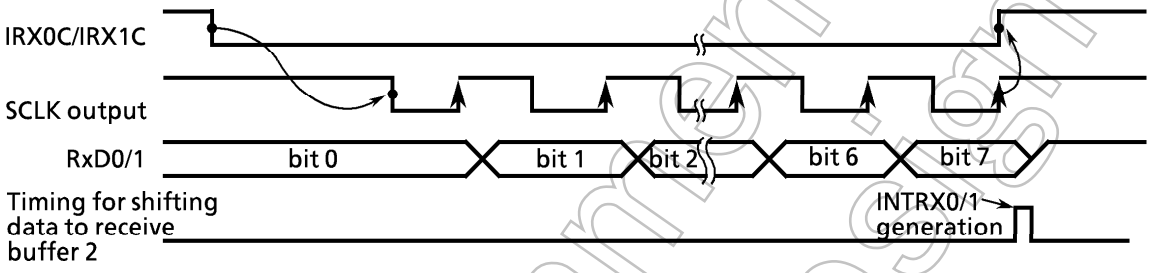


Figure 3.11 (20) Data Receive in I/O Interface Mode (SCLK Output Mode)

In SCLK input mode, if SCLK is input after the CPU reads the received data and clears the receive interrupt flag INTES0<IRX0C>/INTES1<IRX1C>, the next data frame is shifted into receive buffer 1. When an 8-bit data frame is received, the data are shifted to receive buffer 2 (SC0BUF/SC1BUF) and INTES0<IRX0C>/INTES1<IRX1C> is set again, generating interrupt INTRX0/1.

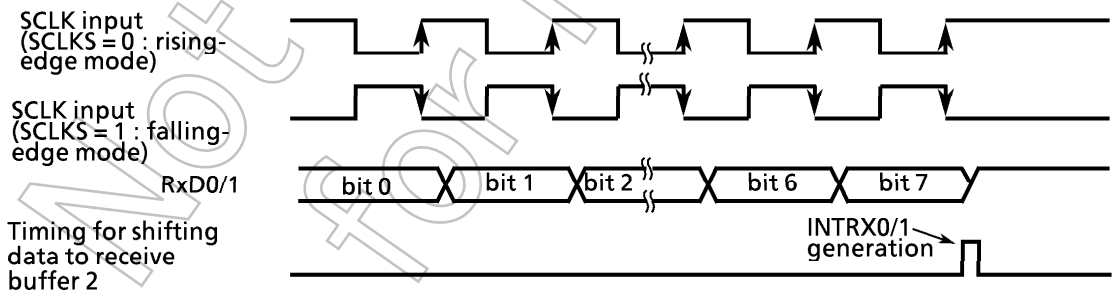


Figure 3.11 (21) Data Receive in I/O Interface Mode (SCLK Input Mode)

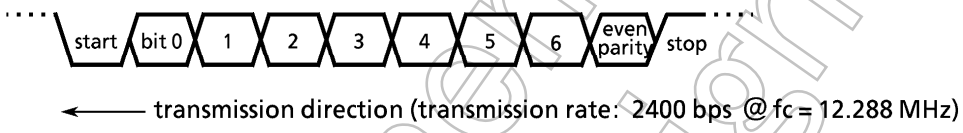
Note : To receive data in either SCLK input mode or SCLK output mode, first enable receive (SC0MOD/SC1MOD<RXE> = "1").

(2) Mode 1 (7-bit UART mode)

Setting the serial channel mode register SC0MOD<SM1,0>/ SC1MOD<SM1,0> to "01" specifies 7-bit UART mode.

A parity bit can be added in this mode. Enable or disable the addition of a parity bit by the serial channel control register SC0CR<PE>/SC1CR<PE> bit. With <PE> set to "1" (parity enabled), select even or odd parity using SC0CR<EVEN>/SC1CR <EVEN>.

Example : When data are transmitted in the following format, the control registers are set as follows. The example shows channel 0.



← transmission direction (transmission rate: 2400 bps @ fc = 12.288 MHz)

	7	6	5	4	3	2	1	0	
PACR	←	-	-	-	-	-	-	-	1	} Sets PA0 as TxD0 pin.
PAFC	←	-	-	X	-	-	-	X	1	
SC0MOD	←	X	0	-	X	0	1	0	1	Sets 7-bit UART mode.
SC0CR	←	X	1	1	X	X	X	0	0	Adds even parity.
BR0CR	←	0	X	1	0	0	1	0	1	Sets transmission rate to 2400bps.
T16RUN	←	1	X	-	-	-	-	-	-	Starts prescaler for baud rate generator.
INTES0	←	1	1	0	0	-	-	-	-	Enables interrupt INTTX0 and sets interrupt level 4.
SC0BUF	←	*	*	*	*	*	*	*	*	Sets transmit data.

Note : X ; Don't care - ; No change

(3) Mode 2 (8-bit UART mode)

Setting serial channel mode register SC0MOD<SM1,0>/SC1MOD<SM1,0> to "10" selects 8-bit UART mode. A parity bit can be added in this mode. Enable or disable the addition of a parity bit by the serial channel control register SC0CR<PE>/SC1CR <PE> bit. With <PE> set to "1" (parity enabled), select even or odd parity using SC0CR<EVEN>/SC1CR<EVEN>.

Example : When data are transmitted in the following format, the control registers are set as follows. The example shows channel 0.

← transmission direction (transmission rate : 9600 bps @ fc = 12.288 MHz)

Main routine settings:

```
          7 6 5 4 3 2 1 0
PACR    ← - - - - - - 0 -     Sets PA1 (RxD0) as input pin.
SC0MOD  ← - 0 1 X 1 0 0 1     Sets 8-bit UART mode and enables reception.
SC0CR   ← X 0 1 X X X 0 0     Adds odd parity.
BR0CR   ← 0 X 0 1 0 1 0 1     Sets transmission rate to 9600bps.
T16RUN  ← 1 X - - - - - -     Starts prescaler for baud rate generator.
INTES0  ← - - - - 1 1 0 0     Enables interrupt INTRX0 and sets interrupt level 4.
```

Interrupt routine processing example:

```
Acc ← SC0CR AND 00011100   }  Checks for errors.
if Acc ≠ 0 then ERROR
Acc ← SC0BUF                  Reads data received.
```

Note: X; Don't care -; No change

(4) Mode 3 (9-bit UART mode)

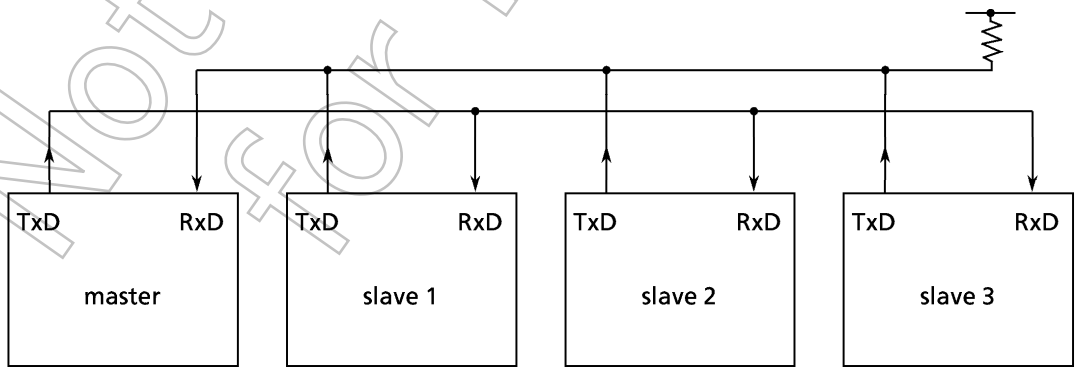
Setting the serial channel mode register SC0MOD<SM1,0>/ SC1MOD<SM1,0> to "11" selects 9-bit UART mode. A parity bit cannot be added in this mode.

At transmission, the most significant bit (9th bit) is written to <TB8> of the serial channel mode register. At receiving, the most significant bit is saved in <RB8> of the serial channel control register.

When data are written to or read from the buffer, the most significant bit is always read or written first, followed by the SC0BUF/SC1BUF register.

Wake-Up Function

In 9-bit UART mode, select the slave controller wake-up function by setting SC0MOD<WU>/SC1MOD<WU>to "1". Interrupt INTRX0/INTRX1 is generated only when <RB8> is set to 1.



Note : Set, in the ODE register, the TxD pin of the slave controller to open drain output mode.

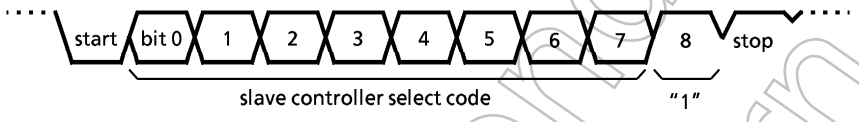
Figure 3.11 (22) Serial Link with Wake-Up Function

Protocol

① Configure the master controller and all slave controllers to 9-bit UART mode.

② Set the SC0MOD<WU>/SC1MOD<WU> bit of each slave controller to "1" to enable data reception.

③ The master controller transmits one frame with the most significant bit (bit 8) <TB8> set to "1". This frame contains the 8-bit select code of a slave controller.



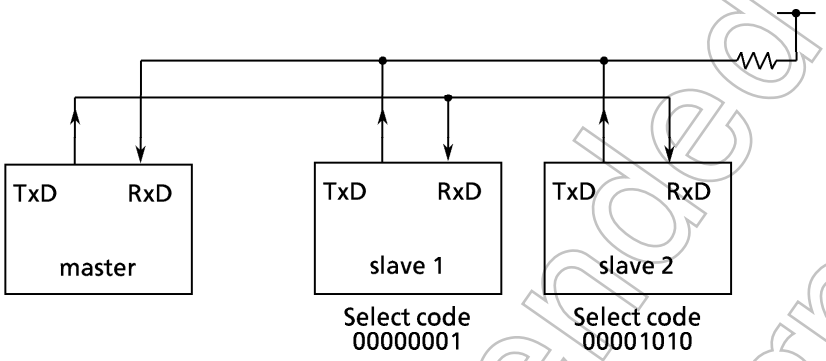
④ The slave controllers receive the above data frame. The slave controller whose select code matches the select code in the data frame received clears its WU bit to 0.

⑤ The master controller transmits data frames with most significant bit (bit 8) <TB8> set to "0" to the specified slave controller (the controller whose SC0MOD<WU>/SC1MOD<WU> bit is cleared to 0).

⑥ The slave controllers not specified (the controllers whose <WU> bit is set to "1") ignore the received data as interrupt INTRX0/INTRX1 is not generated when the most significant bit (bit 8) <RB8> remains cleared to 0 (when data are transmitted).

The specified slave controller (the slave controller whose <WU> bit is set to "0") can transmit data informing the master controller of the termination of a transmission.

Setting example: : When linking two slave controllers serially with the master controller using internal clock $\phi 1$ as the transmission clock.



As serial channels 0 and 1 have the same operation in this mode, the following describes channel 0 only.

- Setting of master controller

Main routine :

```
PACR    ← - - - - - - 0 1   }
PAFC    ← - X X - - X X 1   }  Sets PA0 as TxD pin, and PA1 as RxD pin.
INTES0  ← 1 1 0 0 1 1 0 1   Enables interrupt INTTX0 and sets interrupt level to 4.
                            Enables interrupt INTRX0 and sets interrupt level to 5.
SC0MOD  ← 1 0 1 0 1 1 1 0   Sets to 9-bit UART mode and sets φ1 as transmission clock.
SC0BUF  ← 0 0 0 0 0 0 0 1   Sets select code for slave controller 1.
```

INTTX0 interrupt routine :

```
SC0MOD  ← 0 - - - - - - -   Sets TB8 to 0.
SC0BUF  ← * * * * * * * *   Sets transmit data.
```

- Setting of slave controller 2

Main routine :

```
PACR    ← - - - - - - 0 1   }
PAFC    ← - X X - - X X 1   }  Sets PA0 as TxD pin (open drain output), and PA1 as RxD pin.
ODE     ← X X X X X X - 1   }
INTES0  ← 1 1 0 1 1 1 1 0   Enables INTTX0 and INTRX0.
SC0MOD  ← 0 0 1 1 1 1 1 0   Sets to 9-bit UART mode, sets φ1 2/fc) as transmission clock,
                            and sets <WU> to "1".
```

Interrupt INTRX0 routine :

```
Acc ← SC0BUF
if Acc = select code
Then SC0MOD4 ← - - - 0 - - - -   Clears <WU> to 0.
```

3.12 Analog/Digital Converter

TMP95C063 incorporates a 10-bit successive approximation-type analog/digital converter (A/D converter) with 8-channel analog input.

Figure 3.12 (1) is a block diagram of the A/D converter. The 8-channel analog input pins (AN7 to AN0) are shared by input-only port C and can thus be used as input port pins.

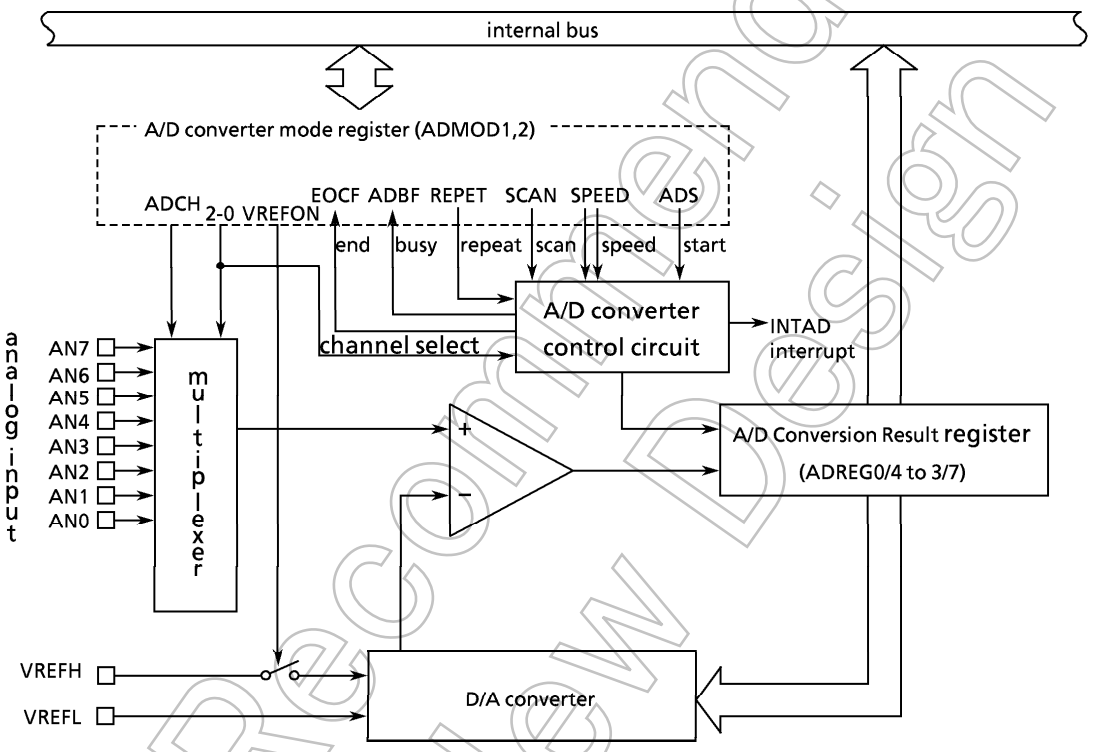


Figure 3.12 (1) Block Diagram of A/D Converter

Note 1 : Because this A/D converter does not include an internal sample and hold circuit, for A/D conversion of high-frequency signals, connect an external sample and hold circuit.

Note 2 : To lower the power supply current in IDLE or STOP mode, depending on the timing, standby mode can be entered with the internal comparator in enable state. Thus, stop A/D conversion before executing the HALT instruction. ADMOD <SPEED1, 2> set the "00".

ADMOD1 (005EH)

	7	6	5	4	3	2	1	0
bit Symbol	EOCF	ADBF	REPET	SCAN		ADS		
Read/Write	R		R/W			R/W		
After reset	0	0	0	0		0		
Function	A/D conversion complete flag 1 : complete	A/D conversion BUSY flag 1 : conversion in progress	Repeat mode specification 0 : single mode 1 : repeat mode	Scan mode specification 0 : conversion channel fixed mode 1 : conversion channel scan mode		A/D conversion start 1 : conversion start Always read as "0".		

A/D conversion start

0	—
1	Starts A/D conversion.

Note : Always read as "0".

A/D scan mode specification

0	A/D conversion channel fixed mode
1	A/D conversion channel scan mode

A/D repeat mode setting

0	A/D single conversion mode
1	A/D repeat conversion mode

A/D conversion BUSY flag

0	A/D conversion halted
1	A/D conversion in progress

A/D conversion complete flag

0	Before or during A/D conversion
1	A/D conversion complete

Figure 3.12 (2-1) A/D Control Register

ADMOD2 (005FH)

	7	6	5	4	3	2	1	0
bit Symbol	VREFON		SPEED1	SPEED0		ADCH2	ADCH1	ADCH0
Read/Write	R/W							
After reset	1		0	0		0	0	0
Function	Ladder resistance ON / OFF		Conversion speed 00 : 160 states 01 : 320 states 10 : 640 states 11 : 1280 states			Analog input channel select		

Analog input channel select

<ADCH2,1,0> \ <SCAN>	0	1
000	AN0	AN0
001	AN1	AN0→AN1
010	AN2	AN0→AN1→AN2
011	AN3	AN0→AN1→AN2→AN3
100	AN4	AN4
101	AN5	AN4→AN5
110	AN6	AN4→AN5→AN6
111	AN7	AN4→AN5→AN6→AN7

A/D conversion speed select

00	160 states (12.8 μs @25 MHz)
01	320 states (25.6 μs @25 MHz)
10	640 states (51.2 μs @25 MHz)
11	1280 states (102.4 μs @25 MHz)

Ladder resistance on/off selection
(See note below)

0	Ladder resistance off
1	Ladder resistance on

Note : Set the <VREFON> bit to "1" before starting conversion (write "1" to ADMOD1<ADS>).

Figure 3.12 (2-2) A/D Control Register

ADREG04L (0060H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR01	ADR00						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Stores lower two bits of AN0 or AN4 conversion result.							

ADREG04H (0061H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight bits of AN0 or AN4 conversion result.							

ADREG15L (0062H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR11	ADR10						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Stores lower two bits of AN1 or AN5 conversion result.							

ADREG15H (0063H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight bits of AN1 or AN5 conversion result.							

Note : Channels AN0 and AN4 share conversion result register ADREG04; AN1 and AN5 share ADREG15; AN2 and AN6 share ADREG26; and AN3 and AN7 share ADREG37.

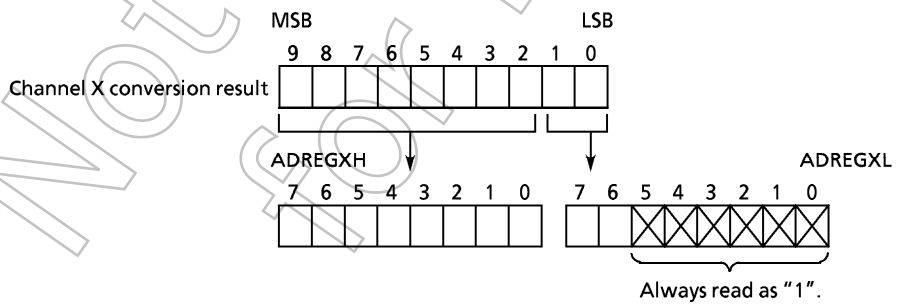


Figure 3.12 (3-1) A/D Conversion Result Registers (ADREG04, 15)

ADREG26L (0064H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Stores lower two bits of AN2 or AN6 conversion result.							

ADREG26H (0065H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight bits of AN2 or AN6 conversion result.							

ADREG37L (0066H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR31	ADR30						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Stores lower two bits of AN3 or AN7 conversion result.							

ADREG37H (0067H)

	7	6	5	4	3	2	1	0
bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight bits of AN3 or AN7 conversion result.							

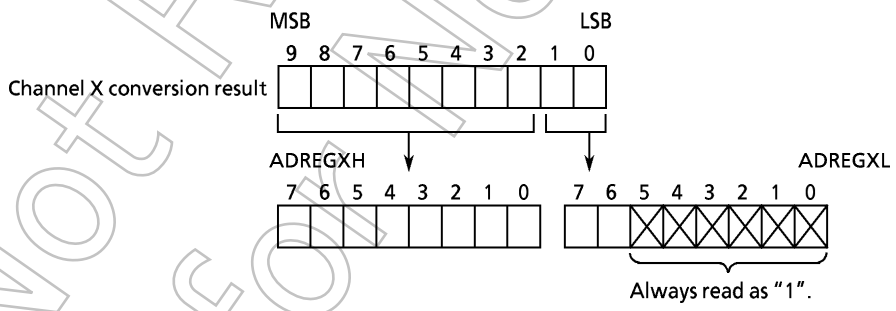


Figure 3.12 (3-2) A/D Conversion Result Registers (ADREG26, 37)

3.12.1 Operation

(1) Analog Reference Voltage

A high analog reference voltage is applied to the VREFH pin; a low analog reference voltage to the VREFL pin.

The reference voltage, the difference between VREFH and VREFL, is divided by 1024 using a ladder resistor. This voltage is compared with the analog input voltage for A/D conversion.

To turn the switch between VREFH and VREFL off, write "0" to the ADMOD2<VREFON> bit.

To start A/D conversion from the off state, first write "1" to <VREFON>, then to ADMOD1<ADS>.

(2) Analog Input Channels

Analog input channel selection depends on the operating mode of the A/D converter.

In analog input channel fixed mode, ADMOD2<ADCH2,1,0> selects one channel among pins AN0 to AN7 (eight pins).

In analog input channel scan mode, ADMOD2<ADCH2,1,0> selects the number of channels to scan. Possible selections are: AN0 only, AN0 → AN1, AN0 → AN1 → AN2, AN0 → AN1 → AN2 → AN3, AN4 only, AN4 → AN5, AN4 → AN5 → AN6, or AN4 → AN5 → AN6 → AN7.

At reset, A/D conversion register ADMOD2<ADCH2,1,0> is initialized to "000", selecting pin AN0 for the A/D converter input.

Pins not used as analog input channels can be used as standard port C input pins.

(3) Starting A/D Conversion

To start A/D conversion, write "1" in A/D conversion start register ADMOD1<ADS>. When A/D conversion starts, A/D conversion BUSY flag ADMOD1<ADBF> is set to "1", indicating A/D conversion is in progress.

Don't set ADMOD1<ADS> to "1" during a conversion. When ADMOD1<ADS> is written "1" during A/D conversion, the conversion is finished halfway and new A/D conversion is started. In the case of conversion channel scan mode, the conversion channel returns to channel 0 and new conversion is started.

(4) A/D Conversion Modes

There are two A/D conversion modes: A/D conversion channel fixed/scan mode and single/repeat conversion mode.

In conversion channel fixed repeat mode, one specified channel is converted repeatedly.

In conversion channel scan repeat mode, the channels are scanned repeatedly. Select the A/D conversion mode using ADMOD1<REPET, SCAN>.

(5) A/D Conversion Speed Selection

There are four A/D conversion speeds. Select the speed using the ADMOD2<SPEED1,0>.

At reset, ADMOD2<SPEED1,0> is initialized to "00", selecting a conversion time of 160 states (12.8 μs @ 25 MHz).

(6) A/D Conversion End and Interrupt

- In A/D single conversion mode

  When A/D conversion ends, the ADMOD1<EOCF> flag is set to "1" to indicate that A/D conversion is complete, the ADMOD1<ADBF> flag is cleared to "0", and an INTAD interrupt is generated. In A/D conversion channel fixed mode, A/D conversion ends when conversion of the specified channel is complete; in A/D conversion channel scan mode, A/D conversion ends when conversion of the final channel is complete.

- In A/D repeat conversion mode

  A/D conversion end interrupt INTAD cannot be used in repeat mode for either conversion channel fixed mode or conversion channel scan mode. Ensure that the INTE0AD register interrupt request level is always set to "000" to disable interrupt requests.

  To end operations in repeat mode, write "0" to the ADMOD1<REPET> register. This ends repeat mode when the conversion currently in progress is complete.

  When A/D conversion changes to the halt state of IDLE and STOP mode, even if in A/D converting state, A/D converter immediately stops the operation. After releasing the halt, the conversion does not restart.

(7) Storing A/D Conversion Result

Channels AN0 and AN4 share conversion result register ADREG04; AN1 and AN5 share ADREG15; AN2 and AN6 share ADREG26; and AN3 and AN7 share ADREG37.

Note that it is impossible to identify the channel whose conversion results are currently being stored in the register. In repeat mode, the registers are updated when each conversion is complete.

Registers ADREG04-37 are read-only.

(8) Reading A/D Conversion Result

Registers ADREG04 to ADRED37 store the A/D conversion result.

Reading any lower 2-bit register ADREGxxL in ADREGxx (xx: 04 - 37), clears INTE0AD<IADC>and ADMOD<EOCF> to 0. Simply reading upper 8-bit register ADREGxxH does not clear <IADC>and<EOCF>.

Setting example : ① Convert the analog input voltage on pin A3 to digital in 160-state mode, and transfer the result to memory address 0100H in the INTAD interrupt routine.

Main routine setting :

```
INTE0AD  ← 1 1 0 0 - - - -     Enables INTAD and sets level 4.
ADMOD2   ← 1 X 0 0 X 0 1 1     Sets analog input channel to pin AN3 and starts
ADMOD1   ← X X 0 0 X 1 X X     A/D conversion in 160-state mode.
```

Interrupt routine processing example :

```
WA          ← ADREG37          Reads ADREG37L and ADREG37H values to WA
                               (16 bits).
WA        >> 6                 Right-shifts WA six times and zero-fills upper bits.

(000100H)←   WA                Writes contents of WA to memory address 0100H.
```

② Repeatedly convert the analog input voltages on pins AN4, AN5, AN6, and AN7 in 320-state in channel scan repeat mode.

```
INTE0AD  ← 1 0 0 0 - - - -     Disables INTAD.
ADMOD2   ← 1 X 0 1 0 1 1 1     Starts A/D conversion of analog input channels
ADMOD1   ← X X 1 1 X 1 0 0     AN4 to AN7 in scan repeat mode.
```

Note: X ; Don't care - ; No change

3.13 8-Bit Voltage Output-Type D/A Converter

TMP95C063 incorporates a 2-channel, 8-bit resolution D/A converter with the following features.

- R-2R-type 8-bit resolution D/A converter with two internal channels
- Registers DAREG0 and DAREG1 to control the analog voltage output

Figure 3.13 (1) is a block diagram of the D/A converter.

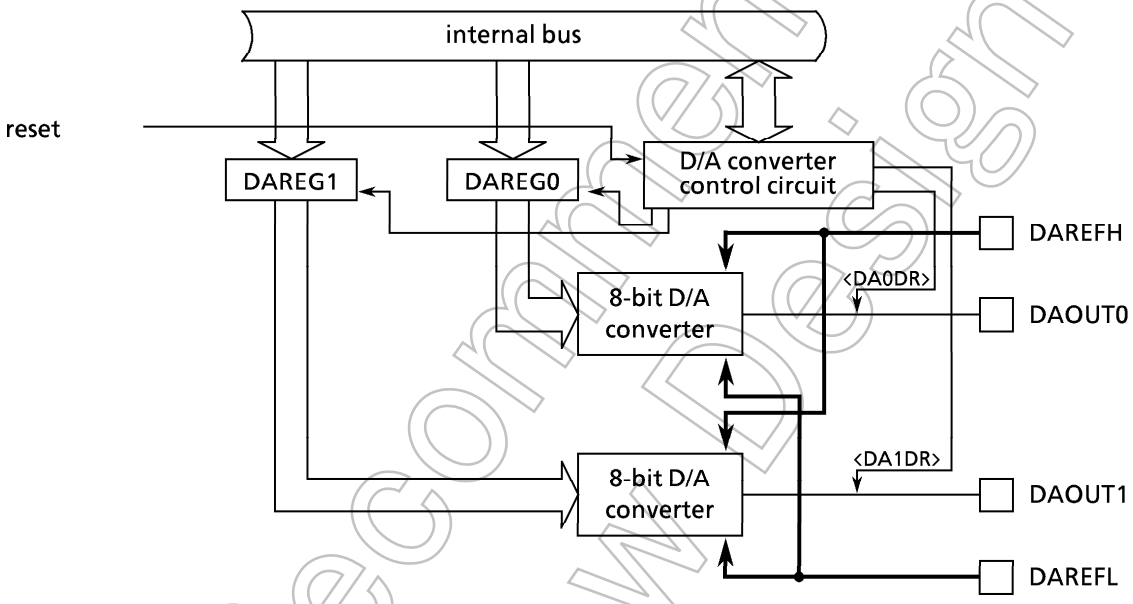


Figure 3.13 (1) D/A Converter Block Diagram

D/A converter drive register

DADRV (004FH)

	7	6	5	4	3	2	1	0
bit Symbol	—	—	—	—	—	—	DA1DR	DA0DR
Read/Write							R/W	
After reset							0	
Function							0 : Outputs 0 V. 1 : Outputs register conversion value.	

D/A output drive setting

0	Outputs 0 V.
1	Outputs register conversion value.

D/A conversion register 0

DAREG0 (004AH)

(No RMW)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	W							
After reset	Undefined							
Function	Starts D/A conversion at register write, outputs to DAOUT0.							

D/A conversion register 1

DAREG1 (004BH)

(No RMW)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	W							
After reset	Undefined							
Function	Starts D/A conversion at a register write, outputs to DAOUT1.							

These registers are used for the D/A converter digital input data. The relationship between the register values and the output voltages is as follows: output voltage V = (DAREFH - DAREFL) x N/256 (where N is the register value).

Note 1: If the HALT instruction is released, the D/A converter outputs the pre-HALT voltage. In HALT modes other than STOP, the D/A converter continues to output the conversion voltage as specified by the value in the registers regardless of the HALT instruction.

Note 2: Read-modify-write is prohibited for registers DAREG0, DAREG1.

Figure 3.13 (2) D/A Converter Registers

3.13.1 Operation

When D/A converter drive register DADRV<DA1DR, DA0DR> is set to "1", the internal D/A converter converts digital values in D/A converter registers DAREG1 or DAREG0 to analog values, and outputs these values as voltages from pins DAOUT1 and DAOUT0. Figure 3.13 (2) shows the relationship between input data and output voltage.

As a reset clears <DA1DR> and <DA0DR> to "0", DAOUT1 and DAOUT0 pins output 0V. After a reset, DAREG1 and DAREG0 are undefined. To output the relevant analog value using the D/A converter, write input data in DAREG1 and DAREG0, then write "1" to the DADRV bit of the channel to be used. Be sure to write data to DAREG1 and DAREG0 first. If, after a reset, DADRV is set to "1" before the input data are written to DAREG1 and DAREG0, DAREG1 and DAREG0 are undefined, and the converter outputs undefined analog values.

If the HALT instruction is executed after specifying STOP mode (WDMOD<HALTM1,0> = "01"), the DAOUT0/DAOUT1 pin outputs 0V regardless of the DADRV or DAREG setting.

Example : Set DAREFH = Vcc, DAREFL = GND.

```
                7 6 5 4 3 2 1 0
DAREG1  ← 1 1 1 1 1 1 1 1   Writes FFH.
DAREG0  ← 1 0 0 0 0 0 0 0   Writes 80H.
DADRV   ← X X X X X X 1 1   Outputs DAOUT1/DAOUT0.
DAREG1  ← 1 0 0 0 0 0 0 0   Writes 80H.  Outputs Vcc / 2 to DAOUT1.
DAREG0  ← 1 1 1 1 1 1 1 1   Writes FFH.  Outputs Vcc to DAOUT0.
```

$$DAOUT1 = Vcc \times \frac{255}{256} \fallingdotseq Vcc$$

$$DAOUT0 = Vcc \times \frac{128}{256} = \frac{Vcc}{2}$$

3.14 Watchdog Timer (Runaway Detection Timer)

TMP95C063 incorporates a watchdog timer for detecting runaways.

The watchdog timer (WDT) returns the CPU to its normal state after the watchdog timer detects the start of a CPU malfunction (runaway) due to noise, for example. When the watchdog timer detects a runaway, it generates a non-maskable interrupt to notify the CPU of the runaway and outputs a "0" signal from the watchdog timer out pin ($\overline{\text{WDTOUT}}$) to notify any peripheral devices of the runaway.

Connecting the watchdog timer output to the RESET pin (within the chip) forces a reset.

3.14.1 Configuration

Figure 3.14 (1) is a block diagram of the watchdog timer (WDT).

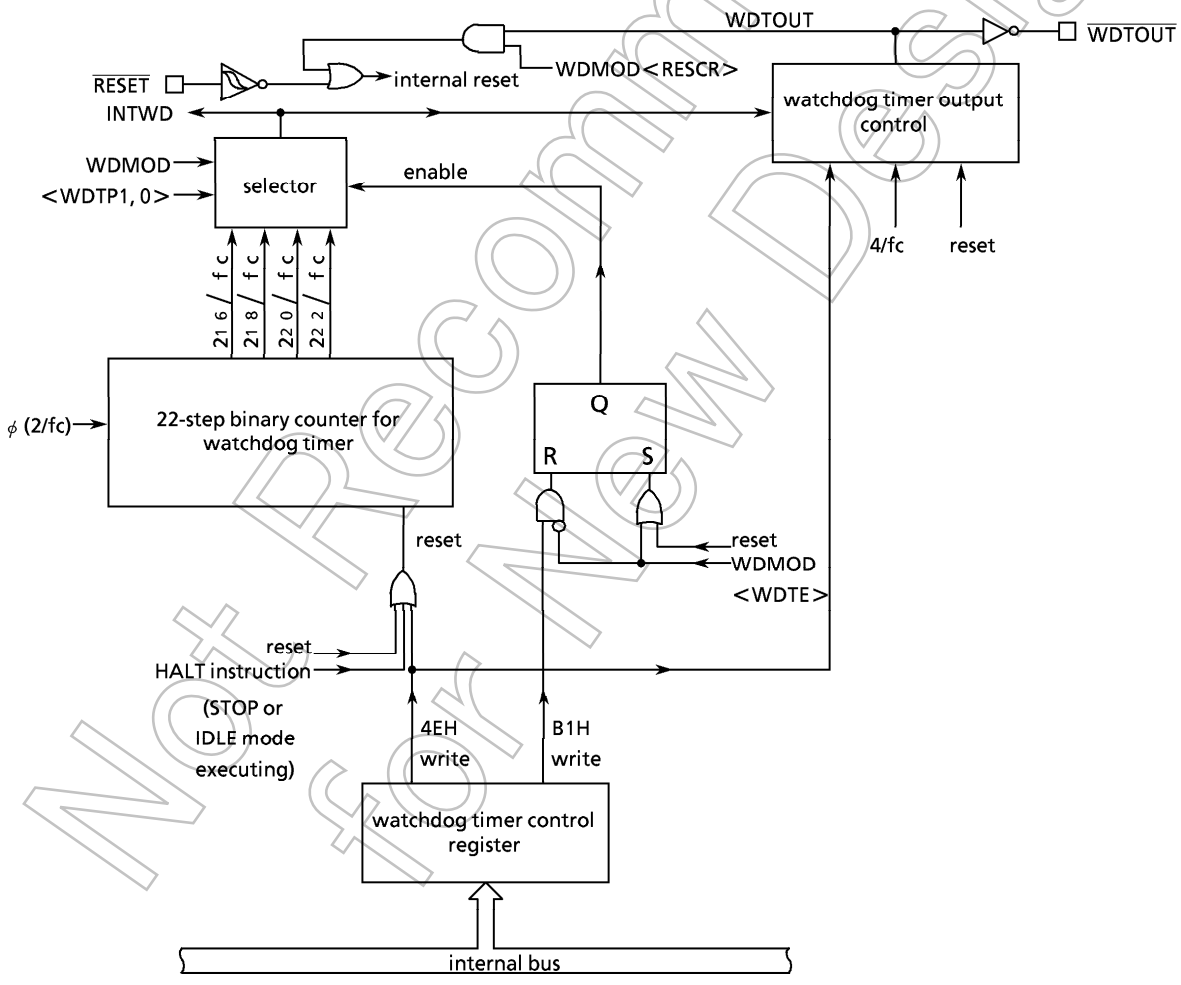


Figure 3.14 (1) Watchdog Timer Block Diagram

The watchdog timer is a 22-step binary counter, which uses ϕ (2/fc) as the input clock.

The WDMOD register selects the output of one of four binary counters: 2^{16}/fc, 2^{18}/fc, 2^{20}/fc, or 2^{22}/fc. Overflow from the selected counter generates a watchdog timer interrupt and outputs a signal to the watchdog timer out pin.

As a result of watchdog timer overflow, the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "0", which can be used as a reset signal for peripheral devices.

First disabling and then clearing the watchdog timer (writing the clear code (4EH) to the WDCR register) sets the WDTOUT pin to "1". In normal mode, the $\overline{\text{WDOUT}}$ pin continually outputs "0" until the clear code is written to the WDCR register.

The watchdog timer output can also be connected to the RESET pin internally. In this case, the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "0" for 8-20 states (640 ns - 1.6 μs @ 25 MHz), resetting itself at the same time.

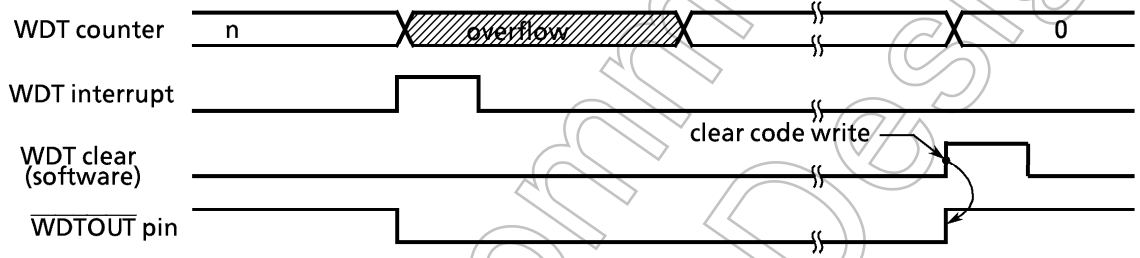


Figure 3.14 (2) Normal Mode

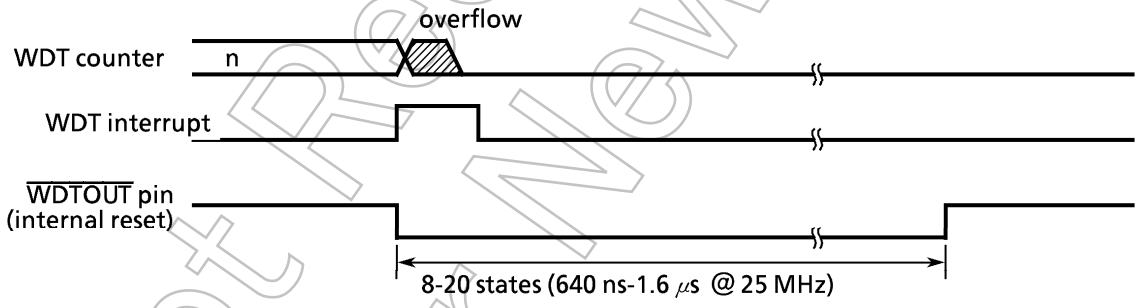


Figure 3.14 (3) Reset Mode

3.14.2 Control Registers

The watchdog timer (WDT) is controlled by two control registers: WDMOD and WDCR.

(1) Watchdog Timer Mode Register WDMOD

① Setting watchdog timer detection time <WDTP>

This 2-bit register is used to set the watchdog timer interrupt time for detecting runaways. At reset, this register is initialized to "00" (WDMOD<WDTP1,0> is set to "00"), setting a detection time of 2^{16}/fc [s]. (The number of states is approximately 32,768.)

② Watchdog timer enable/disable control <WDTE>

At reset, the WDMOD<WDTE> bit is initialized to "1", enabling the watchdog timer.

Disabling the watchdog timer requires both clearing WDTE to 0 and writing disable code B1H in the WDCR register. This two-step process makes it difficult for a runaway to disable the watchdog timer.

To return from the disable state to the enable state, simply set the <WDTE> bit to "1".

③ Connection of watchdog timer output to reset pin <RESCR>

This register determines whether or not the watchdog timer resets itself after a runaway is detected.

At reset, WDMOD<RESCR> is initialized to 0, and the watchdog timer will therefore not trigger a reset.

(2) Watchdog Timer Control Register WDCR

This register is used to disable the watchdog timer functions and to clear the binary counter.

- Disable control

After clearing the WDMOD<WDTE> register to 0, writing the disable code "B1H" to the WDCR register disables the watchdog timer. However, the binary counter continues its operation also after the watchdog timer was disabled.

```
WDMOD ← 0 - - - - - X X      Clears WDTE to 0.
WDCR  ← 1 0 1 1 0 0 0 1      Writes disable code B1H.
```

- Enable control

  Set WDMOD<WDTE> to 1.

  Clear the binary counter before setting the watchdog timer enable. The binary counter continues to count up also after setting the watchdog timer disable, so if the watchdog timer is set enable without clearing the binary counter, the watchdog timer out ($\overline{\text{WDTOUT}}$) signal is output at a different timing from the detecting time which is selected by WDMOD<WDTP1, 0> register.

- Clear control

  Writing clear code 4EH to the WDCR register clears the binary counter and resumes the count.

```
WDCR ← 0 1 0 0 1 1 1 0    Writes clear code 4EH.
```

  The binary counter is cleared when the clear code is written, when reset, and when the device enters standby state in IDLE or STOP mode by execution of the HALT instruction.

  In the case of using the watchdog timer as an interval timer, clear the binary counter in the watchdog timer interrupt sequence. If the binary counter is not cleared in the interrupt sequence, it is cleared by an overflow after it counted up until 22-stage.

WDMOD (006EH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	WDT control 1: enable	WDT detection time select 00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Warm-up time 0: 2^{14}/fc 1: 2^{16}/fc	Standby mode 00: RUN mode 01: STOP mode 10: IDLE mode 11: Don't care		1: Internally connects WDTOUT pin to RESET pin.	1: Drives pins in STOP mode

DREVE (See description of STOP mode in the section on standby function.)

Controls watchdog timer output.

0	—
1	Connects WDTOUT pin to reset pin.

Selects standby mode by HALT instruction

00	RUN mode (halts CPU only)
01	STOP mode (all circuits shut down)
10	IDLE mode (oscillator only in operation)
11	Don't care

Selects warm-up time on return from STOP mode.

0	2^{14}/fc (approximately 655 μs @ 25 MHz)
1	2^{16}/fc (approximately 2.6 ms @ 25 MHz)

Selects watchdog timer detection time.

00	2^{16}/fc (approximately 2.6 ms @ 25 MHz)
01	2^{18}/fc (approximately 10.5 ms @25 MHz)
10	2^{20}/fc (approximately 41.9 ms @25 MHz)
11	2^{22}/fc (approximately 168 ms @25 MHz)

Controls watchdog timer enable/disable.

0	Disable
1	Enable

Figure 3.14 (4) Watchdog Timer Mode Register

WDCR (006FH)

	7	6	5	4	3	2	1	0
bit Symbol	–							
Read/Write	W							
After reset	–							
Function	B1H : WDT disable code 4EH : WDT clear code							

→ Disables and clears WDT.

B1H	Disable code
4EH	Clear code
Others	—

Figure 3.14 (5) Watchdog Timer Control Register

3.14.3 Operation

After the detection time set by the WDMOD<WDTP1,0> register is reached, the watchdog timer generates interrupt INTWD and outputs a low signal to the watchdog timer out pin $\overline{\text{WDTOUT}}$. The binary counter for the watchdog timer must be cleared to 0 by software (instruction) before INTWD is generated. If the CPU malfunctions (runaway) due to causes such as noise and does not execute an instruction to clear the binary counter, the binary counter overflows and generates INTWD.

The CPU interprets INTWD as a malfunction detection signal, which can be used to start the malfunction recovery program to return the system to normal. A CPU malfunction can also be fixed by connecting the watchdog timer output to a reset pin for peripheral devices.

The watchdog timer begins operating immediately on release of the watchdog timer reset.

The watchdog timer is reset and halted in IDLE and STOP modes. The watchdog counter continues counting during bus release ($\overline{\text{BUSAK}}$ = low).

The watchdog timer operates in RUN mode; it can be disabled when RUN mode is entered.

Examples :

① Clear the binary counter.

```
WDCR  ← 0 1 0 0 1 1 1 0    Writes clear code (4EH).
```

② Set the watchdog timer detection time to 2^{18}/fc.

```
WDMOD ← 1 0 1 – – – X X
```

③ Disable the watchdog timer.

```
WDMOD ← 0 – – – – – X X    Clears WDTE to "0".
WDCR  ← 1 0 1 1 0 0 0 1    Writes disable code (B1H).
```

④ Select IDLE mode.

```
WDMOD ← 0 – – – 1 0 X X    Disables WDT and set IDLE mode.
WDCR  ← 1 0 1 1 0 0 0 1
Executes HALT instruction.      Sets to standby mode.
```

⑤ Select STOP mode. (Warm-up time 2^{16}/fc)

```
WDMOD ← – – – 1 0 1 X X    Sets to STOP mode.
Executes HALT instruction.      Sets to standby mode.
```

Note: X ; Don't care – ; No change

3.15 Bus Release Function

TMP95C063 has a bus request pin ($\overline{BUSRQ}$, also used as P53) for releasing the bus, and a bus acknowledge pin ($\overline{BUSAK}$, also used as P54). Set these pins using P5CR and P5FC.

3.15.1 Operation

When "0" is input to the $\overline{BUSRQ}$ pin, TMP95C063 acknowledges a bus release request. When the current bus cycle ends, TMP95C063 sets the address bus (A23-A0) and the bus control signals ($\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{CS0}$-$\overline{CS3}$) simultaneously to high, sets these signals and the output buffer for the data bus (D15 - D0) to off, and sets the $\overline{BUSAK}$ pin to low, indicating that the bus is released.

For the bus release timing when the DRAM controller is in use, and for the states of the DRAM-dedicated pins, See 3.7.5, Bus Release Mode.

During bus release, TMP95C063 disables all access to the internal I/O registers, although the internal I/O functions are not affected. As the watchdog timer continues to count up during bus release, when using the bus release function, set the runaway detection time in accordance with the bus release time.

3.15.2 Pin States at Bus Release

Table 3.15 lists pin states when the bus is released.

Table 3.15 Pin State at Bus Release

Pin Name	Port Mode	Function Mode
D7 to D0	———	At high impedance
P17 to P10 (D15 to D8)	No change	At high impedance
P27 to P20 (A23 to A16)	No change	At high impedance (first set to high)
A15 to A0 $\overline{RD}$ $\overline{WR}$	———	At high impedance (first set to high)
P52 ($\overline{HWR}$) P55 (R/$\overline{W}$)	No change	Set output buffer off (first set to high). Internal pull-up resistors are added regardless of output latch values.
P64 ($\overline{CS3}$) P57 ($\overline{CS2}$) P60 ($\overline{CS1}$) P56 ($\overline{CS0}$)	No change	At high impedance (first set to high)

For P61 to P63 and P65 to P67, see 3.7.5, Bus Release Mode.

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply voltage	V cc	– 0.5 to 6.5	V
Input voltage	V IN	– 0.5 to Vcc + 0.5	V
Output current (total)	Σ IOL	120	mA
Output current (total)	Σ IOH	– 120	mA
Power dissipation (Ta = 70°C)	P D	600	mW
Soldering temperature (10 s)	T SOLDER	260	°C
Storage temperature	T STG	– 65 to 150	°C
Operating temperature	T OPR	– 20 to 70	°C

Note : The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

4.2 DC Electrical Characteristics

Vcc = 5 V ± 10%, TA = – 20 to 70°C (8 to 25 MHz)

(Typ values are for Ta = 25°C and Vcc = 5 V)

Parameter	Symbol	Test Condition	Min	Max	Unit
Input Low Voltage (D0 to 15) P5, P7, P8, P9, PA, PB, PC, PD, PE RESET, NMI, INT0 to 3, INT8, NMI2 EA, AM8/16 X1	V IL V IL1 V IL2 V IL3 V IL4		–0.3 –0.3 –0.3 –0.3 –0.3	0.8 0.3 Vcc 0.25 Vcc 0.3 0.2Vcc	V V V V V
Input High Voltage (D0 to 15) P5, P7, P8, P9, PA, PB, PC, PD, PE RESET, NMI, INT0 to 3, INT8, NMI2 EA, AM8/16 X1	V IH V IH1 V IH2 V IH3 V IH4		2.2 0.7 Vcc 0.75 Vcc Vcc – 0.3 0.8Vcc	Vcc + 0.3 Vcc + 0.3 Vcc + 0.3 Vcc + 0.3 Vcc + 0.3	V V V V V
Output Low Voltage	V OL	I OL = 1.6 mA		0.45	V
Output High Voltage	V OH V OH1 V OH2	I OH = – 400 μA I OH = – 100 μA I OH = – 20 μA	2.4 0.75 Vcc 0.9 Vcc		V V V
Darlington Drive Current (8 Output Pins max.)	I DAR	V EXT = 1.5 V R EXT = 1.1 kΩ	–1.0	– 3.5	mA
Input Leakage Current Output Leakage Current	I LI I LO	0.0 ≦ Vin ≦ Vcc 0.2 ≦ Vin ≦ Vcc – 0.2	0.02 (Typ) 0.05 (Typ)	± 5 ± 10	μA μA
Operating Current (RUN) IDLE STOP (Ta = – 20 to 70°C) STOP (Ta = 0 to 50°C)	I cc	fc = 25 MHz 0.2 ≦ Vin ≦ Vcc – 0.2 0.2 ≦ Vin ≦ Vcc – 0.2	37 (Typ) 3.5 (Typ) 0.5 (Typ)	50 10 50 10	mA mA μA μA
Power Down Voltage (at STOP)	V STOP	V IL2 = 0.2 Vcc, V IH2 = 0.8 Vcc	2.0	6.0	V
RESET Pull Up Resistance	R RST		50	150	kΩ
Pin Capacitance	C IO	fc = 1 MHz		10	pF
Schmitt Width RESET, NMI, INT0 to 3, INT8, NMI2	V TH		0.4	1.0 (Typ)	V
PullUp Resistance	R K		30	150	kΩ

Note: I DAR guarantees driving of up to eight output port pins between any two Vcc pins.

4.3 AC Electrical Characteristics

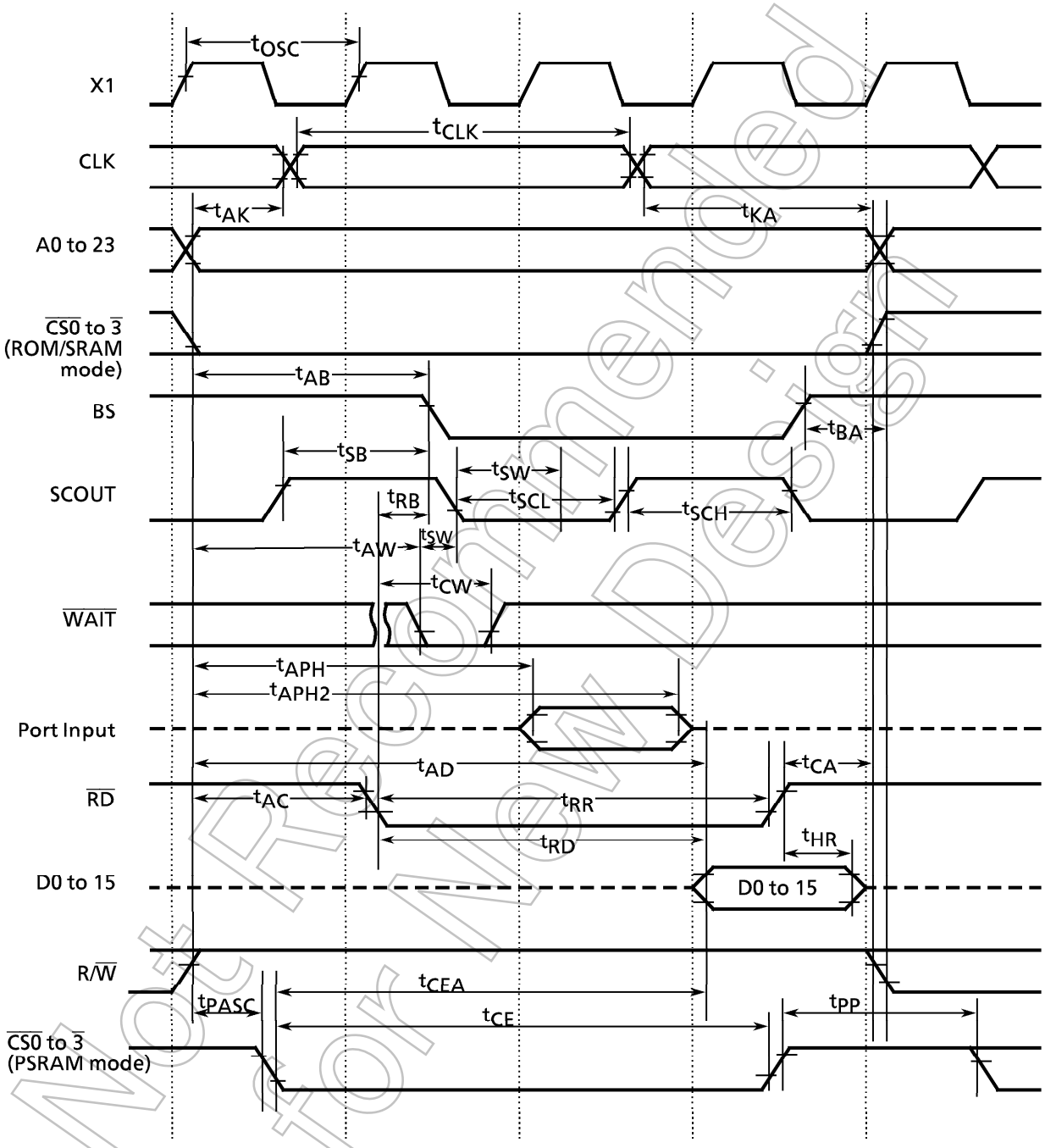
Vcc = 5 V ± 10%, TA = – 20 to 70°C
(8 MHz to 25 MHz)

No.	Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Oscillation cycle (= x)	t_{OSC}	40	125	50		40		ns
2	Clock pulse width	t_{CLK}	2x – 40		60		40		ns
3	A0 to A23 valid→ clock hold	t_{AK}	0.5x – 20		5		0		ns
4	Clock valid→ A0 to A23 hold	t_{KA}	1.5x – 60		5		0		ns
5	A0 to A23 valid→ $\overline{RD}$/$\overline{WR}$ fall	t_{AC}	1.0x – 20		30		20		ns
6	$\overline{RD}$/$\overline{WR}$ rise→ A0 to A23 hold	t_{CA}	0.5x – 20		5		0		ns
7	A0 to A23 valid→ D0 to D15 input	t_{AD}		3.5x – 40		135		100	ns
8	$\overline{RD}$ fall→ D0 to D15 input	t_{RD}		2.5x – 45		80		55	ns
9	$\overline{RD}$ Low pulse width	t_{RR}	2.5x – 40		85		60		ns
10	$\overline{RD}$ rise→ D0 to D15 hold	t_{HR}	0		0		0		ns
11	$\overline{WR}$ Low pulse width	t_{WW}	2.5x – 40		85		60		ns
12	D0 to D15 valid→ $\overline{WR}$ rise	t_{DW}	2.0x – 40		60		40		ns
13	$\overline{WR}$ rise →D0 to D15 hold	t_{WD}	0.5x – 10		15		10		ns
14	A0 to A23 valid→ $\overline{WAIT}$ input (1 + N WAIT)	t_{AW}		3.5x – 90		85		50	ns
	A0 to A23 valid→ $\overline{WAIT}$ input (0 + N WAIT)	t_{AW}		1.5x – 40		35		20	ns
15	$\overline{RD}$/$\overline{WR}$ fall→$\overline{WAIT}$ hold (1 + N WAIT)	t_{CW}	2.5x + 0		125		100		ns
	$\overline{RD}$/$\overline{WR}$ fall→$\overline{WAIT}$ hold (0 + N WAIT)	t_{CW}	0.5x + 0		25		20		ns
16	A0 to 23 valid→ PORT input	t_{APH}		2.5x – 90		35		10	ns
17	A0 to 23 valid→ PORT holt	t_{APH2}	2.5x + 50		175		150		ns
18	$\overline{WR}$ rise→ PORT valid	t_{CP}		200		200		200	ns
19	$\overline{CS}$ Low pulse width (PSRAM mode)	t_{CE}	3x – 40		110		80		ns
20	$\overline{CS}$fall→ D0 to 15 input (PSRAM mood)	t_{CEA}		3x – 60		90		60	ns
21	Address setup time (PSRAM mode)	t_{PASC}	0.5x – 15		10		5		ns
22	$\overline{CS}$ precharge time (PSRAM mode)	t_{PP}	1x – 10		40		30		ns
23	$\overline{RD}$/$\overline{WR}$ fall → BS fall	t_{RB}	0.5x		25		20		ns
24	A0 to A23 valid → BS fall	t_{AB}	1.5x – 20		55		40		ns
25	BS rise → A0 to A23 valid	t_{BA}		0.5x + 15		40		35	ns
26	SCOUT rise → BS fall	t_{SB}		1x + 30		80		70	ns
27	SCOUT fall → $\overline{WAIT}$ input (1 + N WAIT) / (0 + N WAIT)	t_{SW}	60		60		60		ns
28	SCOUT Low pulse width	t_{SCL}	1x – 20		30		20		ns
29	SCOUT High pulse width	t_{SCH}	1x – 20		30		20		ns

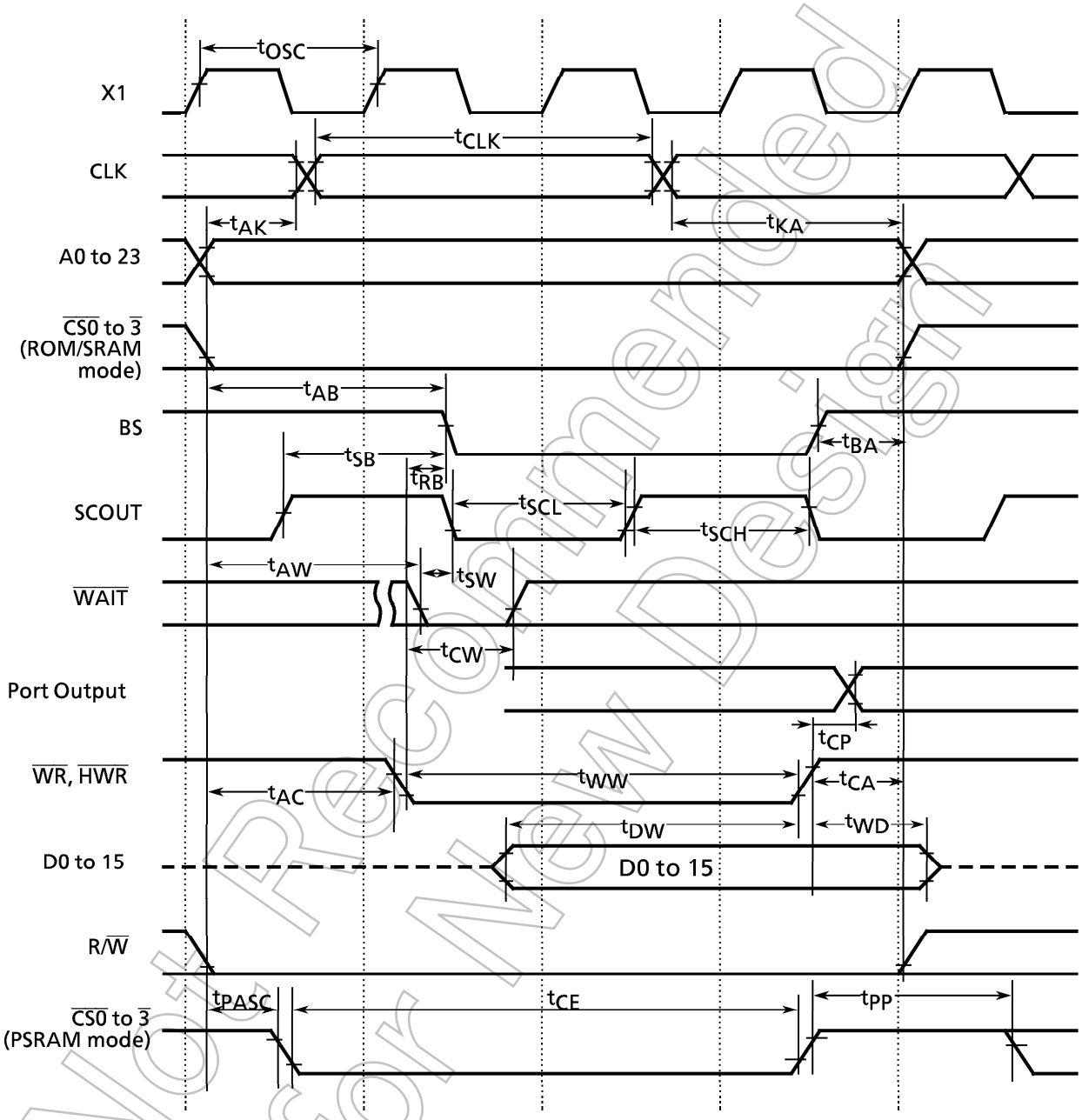
AC measuring conditions

- Output level : High 2.2 V /Low 0.8 V , CL = 50 pF
  (Note that for D0 to D15, A0 to A23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, and CLK, CL = 100 pF, and for SCOUT, CL = 30pF)
- Input level : High 2.4 V /Low 0.45 V (D0 to D15)
  High 0.8 Vcc /Low 0.2 Vcc (except for D0 to D15)

(1) Read cycle



(2) Write cycle



4.4 DRAM Control AC Electrical Characteristics

Vcc = 5 V ± 10%, TA = – 20 to 70°C
(8 MHz to 25 MHz)

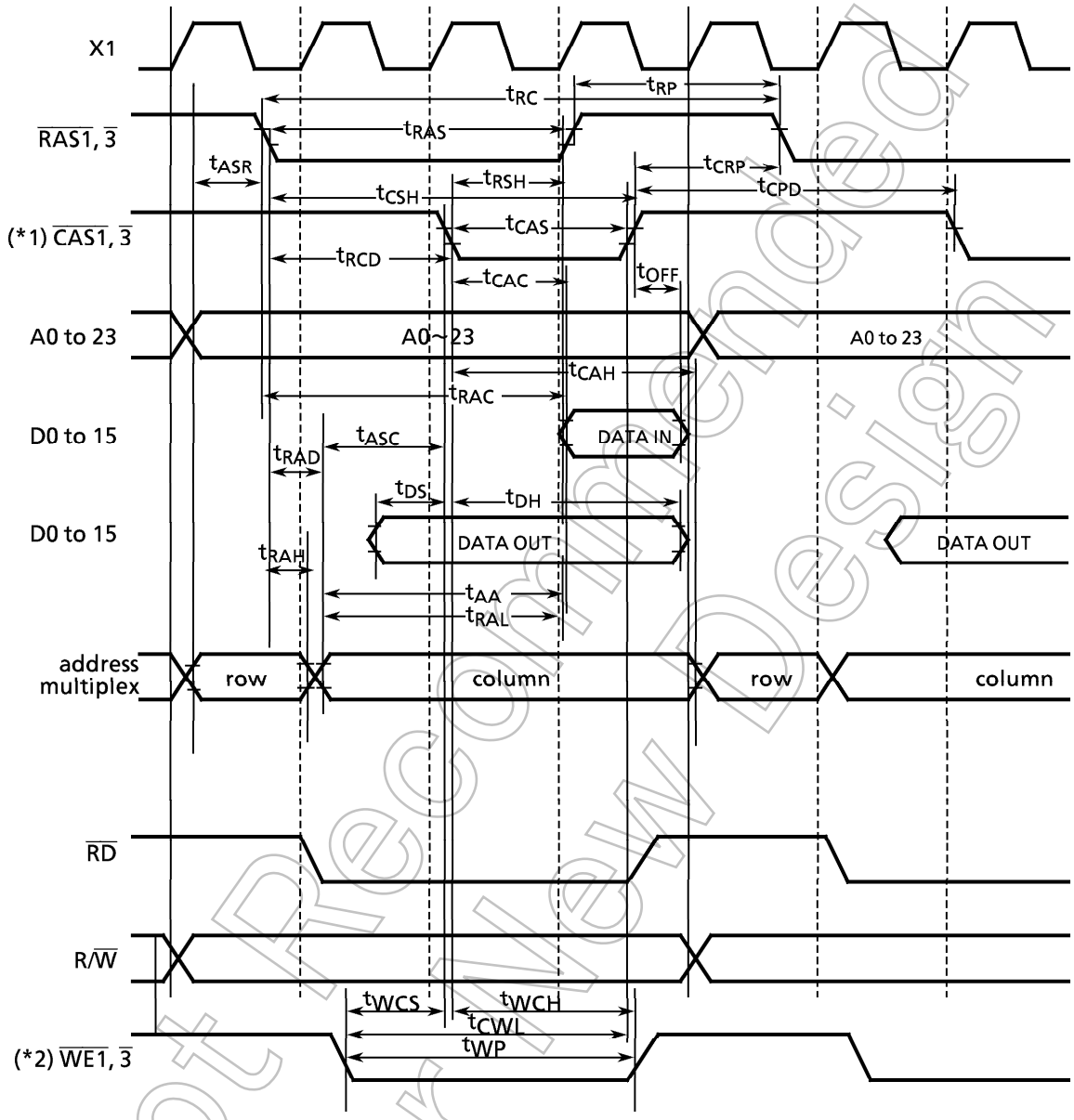
No.	Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	$\overline{RAS}$ cycle time	t_{RC}	4X		200		160		ns
2	$\overline{RAS}$ access time	t_{RAC}		3X-50		100		70	ns
3	$\overline{CAS}$ access time	t_{CAC}		1.5X-35		40		25	ns
4	Column address access time	t_{AA}		2.5X-55		70		45	ns
5	Input data hold time	t_{OFF}	0		0		0		ns
6	$\overline{RAS}$ precharge time	t_{RP}	1.5X-10		65		50		ns
7	$\overline{RAS}$ pulse width	t_{RAS}	2.5X-30		95		70		ns
8	$\overline{RAS}$ hold time	t_{RSH}	1X-15		35		25		ns
9	$\overline{CAS}$ hold time	t_{CSH}	3X-35		115		85		ns
10	$\overline{CAS}$ pulse width	t_{CAS}	1.5X-15		65		45		ns
11	$\overline{RAS}$ to $\overline{CAS}$ delay time	t_{RCD}	1.5X-40	1.5X	35	75	20	60	ns
12	$\overline{RAS}$ column address delay time	t_{RAD}	0.5X-5	0.5X + 20	20	45	15	40	ns
13	$\overline{CAS}$ to $\overline{RAS}$ precharge time	t_{CRP}	1X-35		15		5		ns
14	$\overline{CAS}$ precharge time	t_{CPD}	2.5X-35		90		65		ns
15	Row address setup time	t_{ASR}	0.5X-15		10		5		ns
16	Row address hold time	t_{RAH}	0.5X-5		20		15		ns
17	Column address setup time	t_{ASC}	1X-25		25		15		ns
18	Column address hold time	t_{CAH}	2X-50		50		30		ns
19	Column address $\overline{RAS}$ read time	t_{RAL}	2X-45		55		35		ns
20	Write command $\overline{CAS}$ read time	t_{CWL}	2.0X-35		65		45		ns
21	Data output setup time	t_{DS}	0.5X-15		10		5		ns
22	Data output hold time	t_{DH}	2X-35		65		45		ns
23	Write command setup time	t_{WCS}	0.5X-20		5		0		ns
24	$\overline{CAS}$ hold time ($\overline{CAS}$-before-$\overline{RAS}$)	t_{CHR}*1	2X-50		50		30		ns
25	$\overline{RAS}$ precharge $\overline{CAS}$ active time	t_{RPC}*	1.5X-30		45		30		ns
26	$\overline{CAS}$ setup time ($\overline{CAS}$-before-$\overline{RAS}$)	t_{CSR}*	0.5X-10		15		10		ns
27	$\overline{RAS}$ precharge time (self-refresh)	t_{RPS}*2	4X-20		180		140		ns
28	$\overline{CAS}$ hold time (self-refresh)	t_{CHS}*2	0		0		0		ns
29	Refresh setup time	t_{CFL}*	1X-5		45		35		ns
30	Refresh hold time	t_{CFH}*	1X-10		40		30		ns
31	Write command pulse width	t_{WP}	2.0x-40		60		40		ns
32	Write command hold time	t_{WCH}	1.5x-40		35		20		ns

*1 $\overline{CAS}$-before-$\overline{RAS}$ interval refresh mode
*2 $\overline{CAS}$-before-$\overline{RAS}$ self-refresh mode
* $\overline{CAS}$-before-$\overline{RAS}$ interval refresh and self-refresh modes

AC measuring conditions

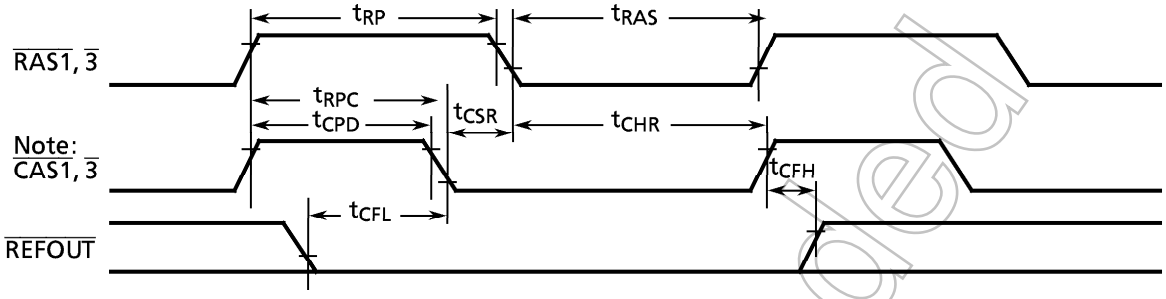
- Output level : High 2.2 V /Low 0.8 V , CL = 50 pF
  (Note that for D0 to D15, A0 to A23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, and R/$\overline{W}$, CL = 100 pF)
- Input level : High 2.4 V /Low 0.45 V (D0 to D15)
  High 0.8 Vcc /Low 0.2 Vcc (except for D0 to D15)

(1) Read/Write Access Cycle

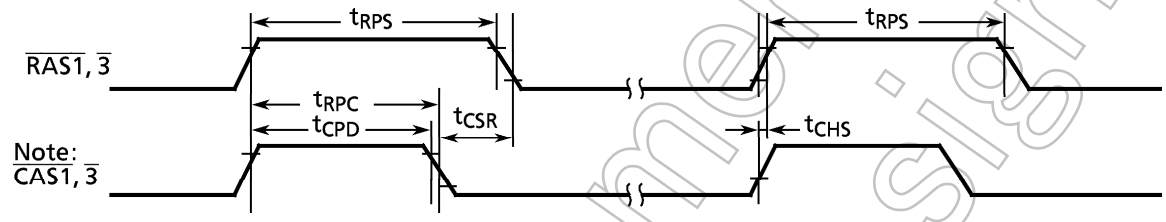


Note 1: Here, $\overline{CAS}$ includes $\overline{LCAS}$ and $\overline{UCAS}$.
Note 2: Here, $\overline{WE}$ includes $\overline{LW}$ and $\overline{UW}$.

(2) $\overline{CAS}$-before-$\overline{RAS}$ Interval Refresh Cycle



(3) $\overline{CAS}$-before-$\overline{RAS}$ Self-Refresh Cycle



Note : Here, $\overline{CAS}$ includes $\overline{LCAS}$ and $\overline{UCAS}$.

4.5 A/D Converter Characteristics

V_{CC} = 5 V ± 10%, TA = − 20 to 70°C (8 to 25 MHz)

Parameter		Symbol	Min	Typ.	Max	Unit
Analog reference voltage (+)		VREFH	V_{CC}-0.2 V	V_{CC}	V_{CC}	V
Analog reference voltage (−)		VREFL	V_{SS}	V_{SS}	V_{SS} + 0.2 V	
Analog input voltage		VAIN	VREFL		VREFH	
Analog reference voltage power supply current		I_{REF}				
V_{CC} = 5 V ± 10%	<VREFON> = 1	(VREFL = 0 V)		0.5	1.5	mA
V_{CC} = 5 V ± 10%	<VREFON> = 0			0.02	5.0	μA
V_{CC} = 5 V ± 10%	Total tolerance	Conversion tolerance		± 3.0	± 6	LSB

Note 1: 1LSB = (VREFH − VREFL) / 2^10 [V]

Note 2: Power supply current ICC from the digital power supply includes the power supply from the AVCC pin.

4.6 Serial Channel Timing

(1) SCLK input mode (I/O interface mode)

V_{CC} = 5 V ± 10%, TA = − 20 to 70°C (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	t_{SCY}	16X		0.8		0.64		μs
Output Data → SCLK rise	t_{OSS}	t_{SCY}/2 − 5X − 50		100		70		ns
SCLK rise→ Output Data hold	t_{OHS}	5X − 100		150		100		ns
SCLK rise→ Input Data hold	t_{HSR}	0		0		0		ns
SCLK rise→ valid data input	t_{SRD}		t_{SCY} − 5X − 100		450		340	ns

(2) SCLK output mode (I/O interface mode) Vcc = 5 V ± 10%, TA = − 20 to 70℃ (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	16X	8192X	0.8	409.6	0.64	327.6	μs
Output Data → SCLK rise	t_{OSS}	t_{SCY} − 2X − 150		550		410		ns
SCLK rise→ Output Data halt	t_{OHS}	2X − 80		20		0		ns
SCLK rise→ Input Data halt	t_{HSR}	0		0		0		ns
SCLK rise→ valid data input	t_{SRD}		t_{SCY} − 2X − 150		550		410	ns

(3) SCLK input mode (UART mode) Vcc = 5 V ± 10%, TA = − 20 to 70℃ (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	t_{SCY}	4X + 20		220		180		ns
SCLK low-level pulse width	t_{SCYL}	2X + 5		105		85		ns
SCLK high-level pulse width	t_{SCYH}	2X + 5		105		85		ns

4.7 Event Counter (TI0, TI2, TI4, TI6, TI8, TI9, TIA, TIB)

Vcc = 5 V ± 10%, TA = − 20 to 70℃ (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock cycle	t_{VCK}	8X + 100		500		420		ns
Clock low-level pulse width	t_{VCKL}	4X + 40		240		200		ns
Clock high-level pulse width	t_{VCKH}	4X + 40		240		200		ns

4.8 Interrupt Operation

Vcc = 5 V ± 10%, TA = − 20 to 70℃ (8 to 25 MHz)

Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, $\overline{NMI2}$, INT0 to 3,8 low-level pulse width	t_{INTAL}	4X		200		160		ns
$\overline{NMI}$, $\overline{NMI2}$, INT0 to 3,8 high-level pulse width	t_{INTAH}	4X		200		160		ns
INT4 to INT7 low-level pulse width	t_{INTBL}	8X + 100		500		420		ns
INT4 to INT7 high-level pulse width	t_{INTBH}	8X + 100		500		420		ns

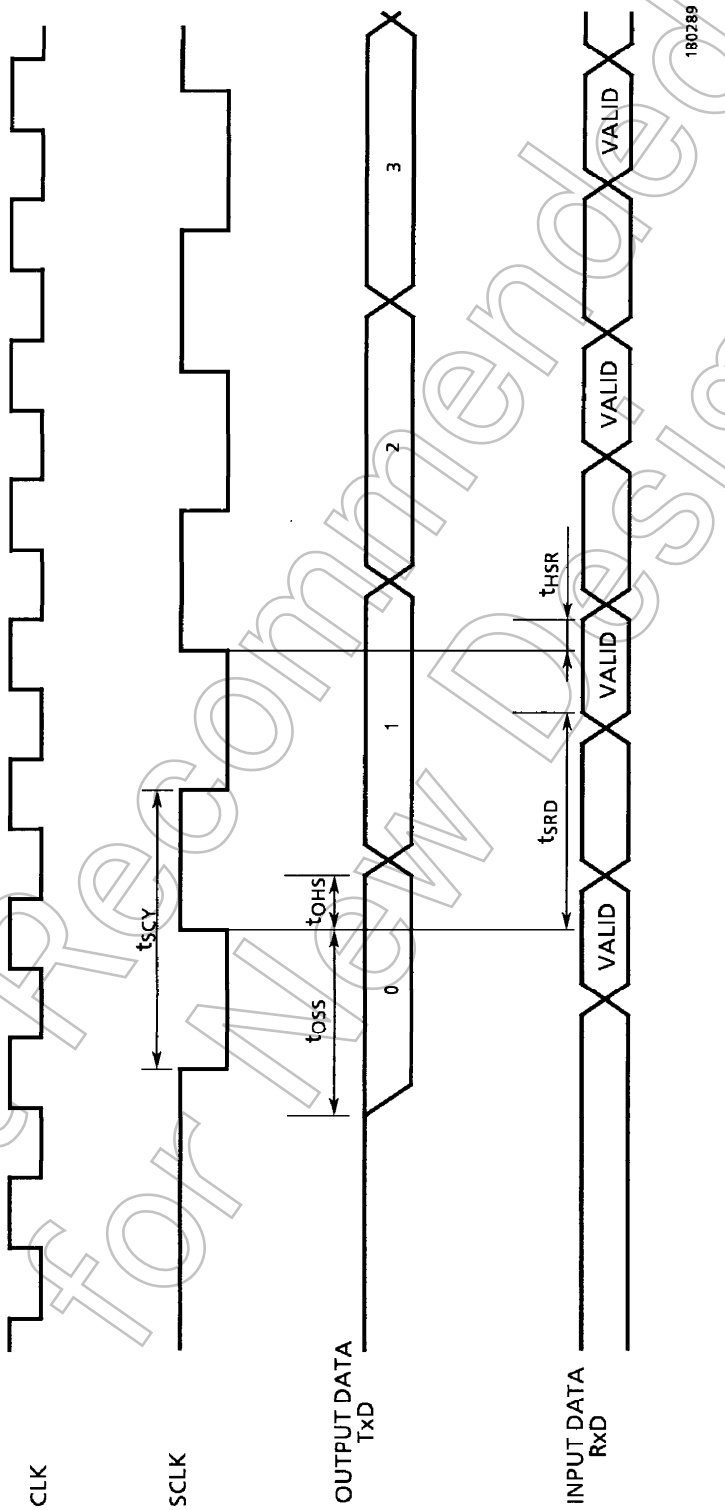
4.9 D/A Conversion Characteristics (Unless otherwise specified, Vcc = 5 V, Vss = DAREFL = 0 V)

Vcc = 5 V ± 10% TA = − 20 to 70℃
f = 8 to 25 MHz

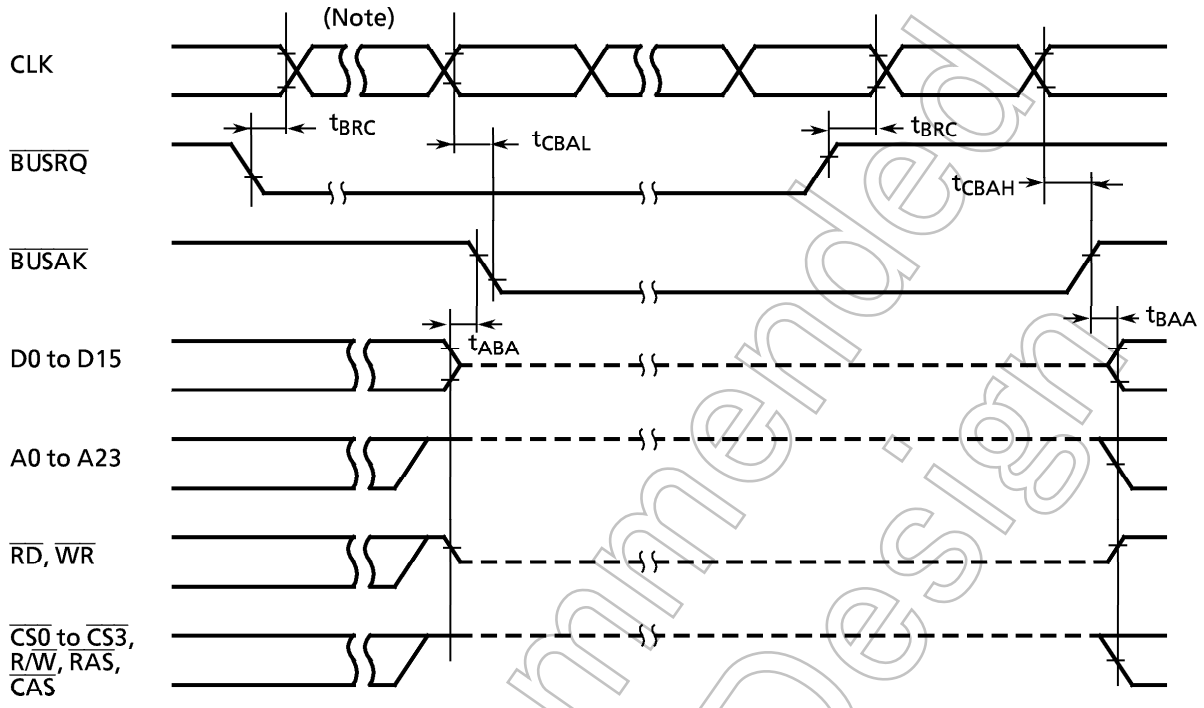
Parameter	Symbol	Condition	Min	Typ	Max	Unit
Analog reference voltage	DAREFH		4.0		Vcc	V
Analog reference voltage	DAREFL		Vss		Vss	
Resolution	Conversion tolerance				8	BIT
Total tolerance		R = 1 MΩ (See note 1) R = 5 MΩ (See note 1) R = 10 MΩ (See note 1)			7.0 4.5 4.0	LSB LSB LSB
Differential linear error				2.0		LSB

Note : "R" is the load resistance on the D/A converter output pin.

4.10 I/O Interface Mode Timing Diagram



4.11 Bus Request/Bus Acknowledge Timing



Parameter	Symbol	Variable		20 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{BUSRQ}$ setup time for CLK	t_{BRC}	120		120		120		ns
CLK→ $\overline{BUSAK}$ fall	t_{CBAL}	2.0x + 120			220		200	ns
CLK→ $\overline{BUSAK}$ rise	t_{CBAH}	0.5x + 40			65		60	ns
Time from output buffer off until $\overline{BUSAK}$ fall	t_{ABA}	0	80	0	80	0	80	ns
Time from $\overline{BUSAK}$ rise until output buffer on	t_{BAA}	0	80	0	80	0	80	ns

Note: When bus release is requested with $\overline{BUSRQ}$ cleared to 0, that request cannot be granted until the previous bus cycle is terminated by a WAIT, and the WAIT is released.

5. List of Special Function Registers

(SFR ; Special Function Register)

The special function registers control the input/output ports and peripheral components. Registers are allocated to 128 bytes within the address range from 000000H to 00007FH.

(1) Input / output port

(2) Input / output port control

(3) Timer control

(4) Pattern generator

(5) Watchdog timer control

(6) Serial channel control

(7) A/D converter control

(8) Interrupt control

(9) Chip select / wait controller

(10) DRAM controller

(11) D/A converter control

Table configuration

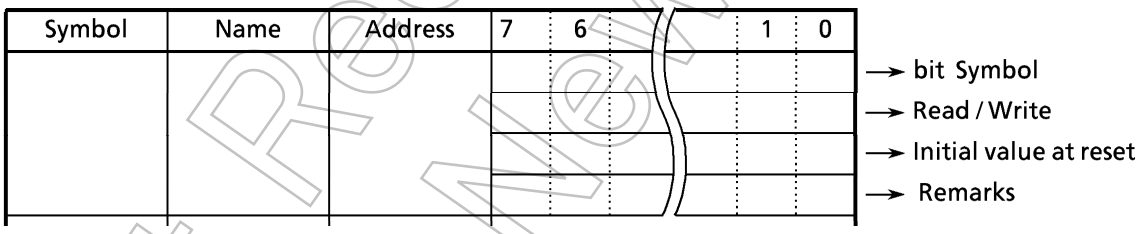


Table 5 I/O Register Address Map

Address	Register Name	Address	Register Name	Address	Register Name	Address	
00000000H		20H	T8RUN	40H	TREGAL	60H	ADREG04L
1H	P1	1H	TRDC	1H	TREGAH	1H	ADREG04H
2H		2H	TREG0	2H	TREGBL	2H	ADREG15L
3H		3H	TREG1	3H	TREGBH	3H	ADREG15H
4H	P1CR	4H	T01MOD	4H	CAP3L	4H	ADREG26L
5H		5H	T02FFCR	5H	CAP3H	5H	ADREG26H
6H	P2	6H	TREG2	6H	CAP4L	6H	ADREG37L
7H		7H	TREG3	7H	CAP4H	7H	ADREG37H
8H		8H	T23MOD	8H	T9MOD	8H	
9H	P2FC	9H	TREG4	9H	T9FFCR	9H	
AH		AH	TREG5	AH	DAREG0	AH	SDMACR0
BH		BH	T45MOD	BH	DAREG1	BH	SDMACR1
CH		CH	T46FFCR	CH	PG0REG	CH	SDMACR2
DH	P5	DH	TREG6	DH	PG1REG	DH	SDMACR3
EH		EH	TREG7	EH	PG01CR	EH	WDMOD
FH		FH	T67MOD	FH	DADRV	FH	WDCR
10H	P5CR	30H	TREG8L	50H	SC0BUF	70H	INTE0AD
1H	P5FC	1H	TREG8H	1H	SC0CR	1H	INTE12
2H	P6	2H	TREG9L	2H	SC0MOD	2H	INTE34
3H	P7	3H	TREG9H	3H	BR0CR	3H	INTE56
4H		4H	CAP1L	4H	SC1BUF	4H	INTE78
5H	P6FC	5H	CAP1H	5H	SC1CR	5H	INTET01
6H	P7CR	6H	CAP2L	6H	SC1MOD	6H	INTET23
7H	P7FC	7H	CAP2H	7H	BR1CR	7H	INTET45
8H	P8	8H	T8MOD	8H	ODE	8H	INTET67
9H	P9	9H	T8FFCR	9H		9H	INTET89
AH	P8CR	AH	T89CR	AH	DMA0V	AH	INTETAB
BH	P8FC	BH	T16RUN	BH	DMA1V	BH	INTES0
CH	P9CR	CH		CH	DMA2V	CH	INTES1
DH	P9FC	DH		DH	DMA3V	DH	INTETC01
EH	PA	EH		EH	ADMOD1	EH	INTETC23
FH	PB	FH		FH	ADMOD2	FH	IIMC
80H	PACR	90H	B0CS				
1H	PAFC	1H	B1CS				
2H	PBCR	2H	B2CS				
3H	PBFC	3H	B3CS				
4H	PC	4H	MSAR0				
5H	PD	5H	MAMR0				
6H		6H	MSAR1				
7H		7H	MAMR1				
8H	PDCR	8H	MSAR2				
9H		9H	MAMR2				
AH	PE	AH	MSAR3				
BH		BH	MAMR3				
CH	PECR	CH	DREFCR1				
DH		DH	DMEMCR1				
EH		EH	DREFCR3				
FH	BEXCS	FH	DMEMCR3				

(1) Input/Output Ports

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	PORT1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Input mode							
			0	0	0	0	0	0	0	0
P2	PORT2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Output mode							
			1	1	1	1	1	1	1	1
P5	PORT5	0DH	P57	P56	P55	P54	P53	P52		RDE
			*R/W							
			Output mode			Input mode (With pull-up)				–
			0	1	1	1	1	1	–	1
P6	PORT6	12H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			Output mode							
			1	1	1	1	1	1	1	1
P7	PORT7	13H	P77	P76	P75	P74	P73	P72	P71	P70
			*R/W							
			Input mode (With pull-up)							
			1	1	1	1	1	1	1	1
P8	PORT8	18H	P87	P86	P85	P84	P83	P82	P81	P80
			*R/W							
			Input mode (With pull-up)							
			1	1	1	1	1	1	1	1
P9	PORT9	19H	P97	P96	P95	P94	P93	P92	P91	P90
			*R/W							
			Input mode (With pull-up)							
			1	1	1	1	1	1	1	1
PA	PORTA	1EH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			*R/W							
			Input mode (With pull-up)							
			1	1	1	1	1	1	1	1
PB	PORTB	1FH	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
			*R/W							
			Input mode (With pull-up)							
			1	1	1	1	1	1	1	1
PC	PORTC	84H	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
			R							
			Input mode							
PD	PORTD	85H				PD4	PD3	PD2	PD1	PD0
						*R/W				
						Input mode (With pull-up)				
						1	1	1	1	1
PE	PORTE	8AH	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
			*R/W							
			Input mode (With pull-up)							
			1	1	1	1	1	1	1	1

Note: When RDE is cleared to 0, the "RD" strobe is output from the "RD" pin (for PSRAM) even when accessing an internal address. If RDE remains set to 1, the "RD" strobe is output only when accessing an external address.

Read/Write

R/W ; Read/Write R ; Read only W ; Write only

No RMW ; Prohibit Read Modify Write. (Cannot use the RES, SET, TSET, CHG, STCF, EX, ADD, ADC, SUB, SBC, INC, DEC, RLC, RRC, RL, RR, SLA, SRA, SLL, SRL, RLD, RRD, AND, OR, or XOR instructions.)

*R/W ; RMW instructions are prohibited for controlling ON/OFF of the pull-up resistors.

(2) Input/Output Port Control (1)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1CR	PORT1 Control	04H	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : IN 1 : OUT							
P2FC	PORT2 Function	09H	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			1	1	1	1	1	1	1	1
		(No RMW)	0 : PORT 1 : A23 to A16							
P5CR	PORT5 Control	10H			P55C	P54C	P53C	P52C		
					W					
					0	0	0	0		
		(No RMW)			0 : IN 1 : OUT					
P5FC	PORT5 Function	11H	P57F	P56F	P55F	P54F	P53F	P52F		
			W							
			0	0	0	0	0	0		
		(No RMW)	0 : PORT 1 : $\overline{CS2}$	0 : PORT 1 : $\overline{CS0}$	0 : PORT 1 : R/$\overline{W}$	0 : PORT 1 : $\overline{BUSAK}$	0 : PORT 1 : $\overline{BUSRQ}$	0 : PORT 1 : $\overline{HWR}$		
P6FC	PORT6 Function	15H	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : PORT 1 : DRAM control signal							
P7CR	PORT7 Control	16H	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : IN 1 : OUT							
P7FC	PORT7 Function	17H	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : PORT		1 : PG1-OUT		0 : PORT		1 : PG0-OUT	
P8CR	PORT8 Control	1AH	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : IN 1 : OUT							
P8FC	PORT8 Function	1BH							P81F	P80F
									W	W
									0	0
		(No RMW)							0 : PORT 1 : SCOUT	0 : PORT 1 : BS
P9CR	PORT9 Control	1CH	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : IN 1 : OUT							
P9FC	PORT9 Function	1DH	P97F		P95F		P93F		P91F	
			W		W		W		W	
			0		0		0		0	
		(No RMW)	0 : PORT 1 : TO7		0 : PORT 1 : TO5		0 : PORT 1 : TO3		0 : PORT 1 : TO1	

Symbol	Name	Address	7	6	5	4	3	2	1	0
PACR	PORTA Control	80H	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : IN 1 : OUT							
PAFC	PORTA Function	81H	PA7F			PA4F	PA3F			PA0F
			W			W	W			W
			0			0	0			0
		(No RMW)	0 : PORT 1 : SCLK1			0 : PORT 1 : TxD1	0 : PORT 1 : SCLK0			0 : PORT 1 : TxD0
PBCR	PORTB Control	82H	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : IN 1 : OUT							
PBFC	PORTB Function	83H	PB7F	PB6F			PB3F	PB2F		
			W	W			W	W		
			0	0			0	0		
		(No RMW)	0 : PORT 1 : TOB	0 : PORT 1 : TOA			0 : PORT 1 : TO9	0 : PORT 1 : TO8		
PDCR	PORTD Control	88H				PD4C	PD3C	PD2C	PD1C	PD0C
						W				
						0	0	0	0	0
		(No RMW)				0 : IN 1 : OUT				
PECR	PORTE Control	8CH	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
			W							
			0	0	0	0	0	0	0	0
		(No RMW)	0 : IN 1 : OUT							

(3) Timer Control (1)

Symbol	Name	Address	7	6	5	4	3	2	1	0
T8RUN	8 bit Timer Control	20H	P7RUN	P6RUN	T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
			R/W							
			0	0	0	0	0	0	0	0
			8 Bit Timer Run/Stop CONTROL 0 : Stop & Clear 1 : Run (Count up)							
TREG0	8 bit Timer Register 0	22H (No RMW)	–							
			W							
			Undefined							
TREG1	8 bit Timer Register 1	23H (No RMW)	–							
			W							
			Undefined							
T01 MOD	8 bit Timer 0,1 Source CLK & MODE	24H (No RMW)	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00 : 8 bit Timer 01 : 16 bit Timer 10 : 8 bit PPG 11 : 8 bit PWM		00 : – 01 : 2^6 – 1 10 : 2^7 – 1 11 : 2^8 – 1 PWM cycle		00 : TO0TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI0 input 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T02 FFCR	8 bit Timer Flip-Flop Control	25H (No RMW)	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
			W		R/W		W		R/W	
			–		0	0	–		0	0
			00 : Invert TFF3 01 : Set TFF3 10 : Clear TFF3 11 : Don't care		1 : TFF3 Invert Enable	1 : Inverted by timer 3	00 : Invert TFF1 01 : Set TFF1 10 : Clear TFF1 11 : Don't care		1 : TFF1 Invert Enable	1 : Inverted by timer 1
TREG2	8 bit Timer Register 2	26H (No RMW)	–							
			W							
			Undefined							
TREG3	8 bit Timer Register 3	27H (No RMW)	–							
			W							
			Undefined							
T23 MOD	8 bit Timer 2,3 Source CLK & MODE	28H (No RMW)	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00 : 8 bit Timer 01 : 16 bit Timer 10 : 8 bit PPG 11 : 8 bit PWM		00 : – 01 : 2^6 – 1PWM 10 : 2^7 – 1 cycle 11 : 2^8 – 1		00 : TO2TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI2 01 : ϕT1 10 : ϕT4 11 : ϕT16	
TRDC	Timer Reg. Double Buffer Control Reg.	21H					TR6DE	TR4DE	TR2DE	TR0DE
							R/W			
							0	0	0	0
								0 : Double Buffer Disable 1 : Double Buffer Enable		

Timer Control (2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG4	8 bit Timer Register 4	29H (No RMW)	–							
			W							
			Undefined							
TREG5	8 bit Timer Register 5	2AH (No RMW)	–							
			W							
			Undefined							
T45 MOD	8 bit Timer 4,5 Source CLK & MODE	2BH (No RMW)	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00 : 8 bit Timer 01 : 16 bit Timer 10 : 8 bit PPG 11 : 8 bit PWM		00 : – 01 : 2^6 – 1 10 : 2^7 – 1 11 : 2^8 – 1 PWM cycle		00 : TO4TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI4 input 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T46 FFCR	8 bit Timer Flip-Flop Control	2CH (No RMW)	TFF7C1	TFF7C0	TFF7IE	TFF7IS	TFF5C1	TFF5C0	TFF5IE	TFF5IS
			W		R/W		W		R/W	
			–		0	0	–		0	0
			00 : Invert TFF7 01 : Set TFF7 10 : Clear TFF7 11 : Don't care		1 : TFF7 Invert Enable	1 : Inverted by timer 7	00 : Invert TFF5 01 : Set TFF5 10 : Clear TFF5 11 : Don't care		1 : TFF5 Invert Enable	1 : Inverted by timer 5
TREG6	8 bit Timer Register 6	2DH (No RMW)	–							
			W							
			Undefined							
TREG7	8 bit Timer Register 7	2EH (No RMW)	–							
			W							
			Undefined							
T67 MOD	8 bit Timer 6,7 Source CLK & MODE	2FH (No RMW)	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00 : 8 bit Timer 01 : 16 bit Timer 10 : 8 bit PPG 11 : 8 bit PWM		00 : – 01 : 2^6 – 1PWM 10 : 2^7 – 1 cycle 11 : 2^8 – 1		00 : TO6TRG 01 : ϕT1 10 : ϕT16 11 : ϕT256		00 : TI6 01 : ϕT1 10 : ϕT4 11 : ϕT16	

Timer Control (3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG8L	16 bit Timer Register8L	30H (No RMW)	–							
			W							
			Undefined							
TREG8H	16 bit Timer Register8H	31H (No RMW)	–							
			W							
			Undefined							
TREG9L	16 bit Timer Register9L	32H (No RMW)	–							
			W							
			Undefined							
TREG9H	16 bit Timer Register9H	33H (No RMW)	–							
			W							
			Undefined							
CAP1L	Capture Register1L	34H	–							
			R							
			Undefined							
CAP1H	Capture Register1H	35H	–							
			R							
			Undefined							
CAP2L	Capture Register2L	36H	–							
			R							
			Undefined							
CAP2H	Capture Register2H	37H	–							
			R							
			Undefined							
T8MOD	16 bit Timer 8 Source CLK & MODE	38H (No RMW)	CAP2T9	EQ9T9	CAP1IN	CAP12M1	CAP12M0	CLE	T8CLK1	T8CLK0
			R/W		W			R/W		
			0	0	1	0	0	0	0	0
			TFF9 INV TRG 0 : TRG Disable 1 : TRG Enable		0 : Soft-Capture 1 : Don't care	Capture Timming 00 : Disable 01 : TI8 ↑ TI9 ↑ 10 : TI8 ↑ TI8 ↓ 11 : TFF1 ↑ TFF1 ↓		1 : UC8 Clear Enable	Source Clock 00 : TI8 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T8FFCR	16 bit Timer 8 Flip-Flop Control	39H (No RMW)	TFF9C1	TFF9C0	CAP2T8	CAP1T8	EQ9T8	EQ8T8	TFF8C1	TFF8C0
			W		R/W				W	
			–		0	0	0	0	–	
			00 : Invert TFF9 01 : Set TFF9 10 : Clear TFF9 11 : Don't care		TFF8 Invert Trigger 0 : Trigger Disable 1 : Trigger Enable				00 : Invert TFF8 01 : Set TFF8 10 : Clear TFF8 11 : Don't care	
T89CR	T8, T9 Control	3AH	–				PG1T	PG0T	DBAEN	DB8EN
			R/W				R/W			
			0				0	0	0	0
			Fixed to "0"				PG1 shift trigger 0 : timer 2, 3 1 : timer 9	PG0 shift trigger 0 : timer 0, 1 1 : timer 8	1 : Double Buffer Enable	
T16RUN	16 bit Timer Control	3BH	PRRUN		T9RUN	T8RUN				
			R/W		R/W	R/W				
			0		0	0				
			Prescaler & 16 Bit Timer Run/Stop control 0 : Stop & Clear 1 : Run (Count Up)							

Timer Control (4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREGAL	16 bit Timer RegisterAL	40H (No RMW)	–							
			W							
			Undefined							
TREGAH	16 bit Timer RegisterAH	41H (No RMW)	–							
			W							
			Undefined							
TREGBL	16 bit Timer RegisterBL	42H (No RMW)	–							
			W							
			Undefined							
TREGBH	16 bit Timer RegisterBH	43H (No RMW)	–							
			W							
			Undefined							
CAP3L	Capture Register3L	44H	–							
			R							
			Undefined							
CAP3H	Capture Register3H	45H	–							
			R							
			Undefined							
CAP4L	Capture Register4L	46H	–							
			R							
			Undefined							
CAP4H	Capture Register4H	47H	–							
			R							
			Undefined							
T9MOD	16 bit Timer 9 Source CLK & MODE	48H (No RMW)	CAP4TB	EQBTB	CAP3IN	CAP34M1	CAP34M0	CLE	T9CLK1	T9CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFFB INV TRG 0 : TRG Disable 1 : TRG Enable		0 : Soft-Capture 1 : Don't care	Capture Timming 00 : Disable 01 : TIA ↑ TIB ↑ 10 : TIA ↑ TIA ↓ 11 : TFF1 ↑ TFF1 ↓		1 : UC9 Clear Enable	Source Clock 00 : TIA 01 : ϕT1 10 : ϕT4 11 : ϕT16	
T9FFCR	16 bit Timer 9 Flip-Flop Control	49H (No RMW)	TFFBC1	TFFBC0	CAP4TA	CAP3TA	EQBTA	EQATA	TFFAC1	TFFAC0
			W		R/W				W	
			–		0	0	0	0	–	
			00 : Invert TFFB 01 : Set TFFB 10 : Clear TFFB 11 : Don't care			TFFA Invert Trigger 0 : Trigger Disable 1 : Trigger Enable			00 : Invert TFFA 01 : Set TFFA 10 : Clear TFFA 11 : Don't care	

(4) Pattern Generator

Symbol	Name	Address	7	6	5	4	3	2	1	0
PG0REG	PG0 Register	4CH (No RMW)	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
			W				R/W			
			0	0	0	0	Undefined			
PG1REG	PG1 Register	4DH (No RMW)	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
			W				R/W			
			0	0	0	0	Undefined			
PG01CR	PG0, 1 Contorol	4EH	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
			R/W							
			0	0	0	0	0	0	0	0
			0: 8 bit write 1: 4 bit write	0: normal rotation 1: reverse rotation	0: 4 bit Step 1: 8 bit Step	PG1 trigger input enable 1: enable	0: 8 bit write 1: 4 bit write	0: normal rotation 1: reverse rotation	0: 4 bit Step 1: 8 bit Step	PG0 trigger input enable 1: enable

(5) Watchdog Timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WD-MOD	Watch Dog Timer Mode	6EH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W							
			1	0	0	0	0	0	0	0
			1: WDT Enable	00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Warming up Time 0: 2^{14}/fc 1: 2^{16}/fc	Standby Mode 00: RUN Mode 01: STOP Mode 10: IDLE Mode 11: Don't care		1: internally connects WDT output to RESET pin.	1: Drives pins even in STOP mode.
WDCR	Watch Dog Timer Control Register	6FH (No RMW)	–							
			W							
			–							
			B1H: WDT Disable Code 4EH: WDT Clear Code							

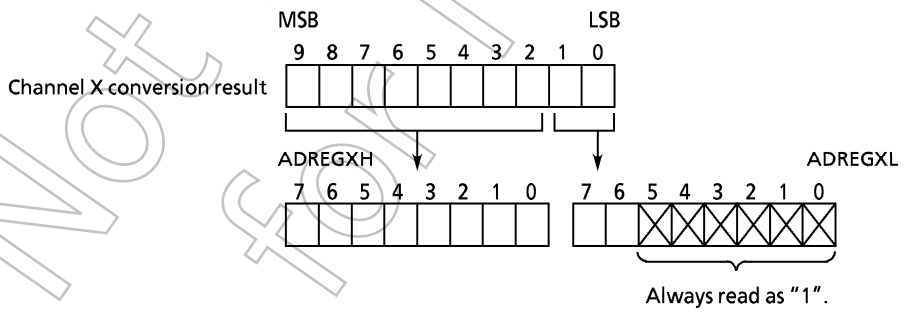
(6) Serial Channels

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer	50H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC0CR	Serial Channel 0 Control	51H	RB8	EVEN	PE	OERR	PERR	FERR	SCLK	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			0	0	0	0	0	0	0	0
			Receive data bit 8	Parity 0: Odd 1: Even	1: Parity Enable	Overrun	1: Error Parity	Framing	0: SCLK0 (↑) 1: SCLK0 (↓)	1: SCLK0 pin input
SC0-MOD	Serial Channel 0 Mode	52H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmit data bit 8	1: CTS Enable	1: Receive Enable	1: Wake up Enable	00: I/O Interface 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TO2 Trigger 01: baud rate generator 10: internal clock ϕ1 11: external clock SCLK0	
BR0CR	Baud Rate Control	53H	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W				R/W			
			0		0	0	0	0	0	0
			Fix to "0"		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Set division value 0 to F			
SC1BUF	Serial Channel 1 Buffen	54H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC1CR	Serial Channel 1 Control	55H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			0	0	0	0	0	0	0	0
			Receive data bit 8	Parity 0: Odd 1: Even	1: Parity Enable	Overrun	1: Error Parity	Framing	0: SCLK1 (↑) 1: SCLK1 (↓)	1: SCLK1 pin input
SC1-MOD	Serial Channel 1 Mode	56H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmit data bit 8	1: CTS Enable	1: Receive Enable	1: Wake up Enable	00: I/O Interface 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TO2 Trigger 01: baud rate generator 10: internal clock ϕ1 11: external clock SCLK1	
BR1CR	Baud Rate Control	57H	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W				R/W			
			0		0	0	0	0	0	0
			Fix to "0"		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Sets divisor. 0 to F			
ODE	Serial Open Drain Enable	58H							ODE1	ODE0
									R/W	
									0	0
									1:P83 open drain	1:P80 open drain

(7) A/D Converter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD 1	A/D Mode Reg 1	5EH	EOCF	ADBF	RPT	SCAN		ADS		
			R		R/W			R/W		
			0	0	0	0		0		
			1: End	1: Busy	1: Repeat	1: Scan		1: Start		
ADMOD 2	A/D Mode Reg 2	5FH	VREFON		SPEED1	SPEED0		ADCH2	ADCH1	ADCH0
			R/W		R/W			R/W		
			1		0	0		0	0	0
			Ladder resistor SW		Speed selection			Input channel selection		
AD REG04L *1)	AD Result Reg 0/4 low	60H	ADR01	ADR00						
			R							
			Undefined		1	1	1	1	1	1
AD REG04H	AD Result Reg 0/4 high	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
AD REG15L *1)	AD Result Reg 1/5 low	62H	ADR11	ADR10						
			R							
			Undefined		1	1	1	1	1	1
AD REG15H	AD Result Reg 1/5 high	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
AD REG26L *1)	AD Result Reg 2/6 low	64H	ADR21	ADR20						
			R							
			Undefined		1	1	1	1	1	1
AD REG26H	AD Result Reg 2/6 high	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
AD REG37L *1)	AD Result Reg 3/7 low	66H	ADR31	ADR30						
			R							
			Undefined		1	1	1	1	1	1
AD REG37H	AD Result Reg 3/7 high	67H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

*1) The data stored in A/D result register low are the lower two bits of the conversion result. The lower six bits of A/D result register low are always read as "1".



(8) Interrupt Control (1)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE-0AD	INTerrupt Enable 0 & A/D	70H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTE12	INTerrupt Enable 2/1	71H	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTE34	INTerrupt Enable 4/3	72H	INT4				INT3			
			I4C	I4M2	I4M1	I4M0	I3C	I3M2	I3M1	I3M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTE56	INTerrupt Enable 6/5	73H	INT6				INT5			
			I6C	I6M2	I6M1	I6M0	I5C	I5M2	I5M1	I5M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTE78	INTerrupt Enable 10/7	74H	INT8				INT7			
			I8C	I8M2	I8M1	I8M0	I7C	I7M2	I7M1	I7M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTET01	INTerrupt Enable Timer 1/0	75H	INTT1 (timer 1)				INTT0 (timer 0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTET23	INTerrupt Enable Timer 3/2	76H	INTT3 (timer 3)				INTT2 (timer 2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	
INTET45	INTerrupt Enable Timer 5/4	77H	INTT5 (timer 5)				INTT4 (timer 4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTET67	INTerrupt Enable Timer 7/6	78H	INTT7 (timer 7)				INTT6 (timer 6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTET89	INTerrupt Enable Timer 9/8	79H	INTTR9 (TREG9)				INTTR8 (TREG8)			
			IT9C	IT9M2	IT9M1	IT9M0	IT8C	IT8M2	IT8M1	IT8M0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0
INTETAB	INTerrupt Enable Timer B/A	7AH	INTTRB (TREGB)				INTTRA (TREGA)			
			ITBC	ITBM2	ITBM1	ITBM0	ITAC	ITAM2	ITAM1	ITAM0
			R/W	W			R/W	W		
		(No RMW)	0	0	0	0	0	0	0	0

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request.
0	0	1	Sets interrupt request level to "1".
0	1	0	Sets interrupt request level to "2".
0	1	1	Sets interrupt request level to "3".
1	0	0	Sets interrupt request level to "4".
1	0	1	Sets interrupt request level to "5".
1	1	0	Sets interrupt request level to "6".
1	1	1	Disables interrupt request.

IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request.	Clears interrupt request flag.
1	Indicates interrupt request.	- - - - - Don't care - - - - -

Interrupt Control (2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTES0	INTerrupt Enable Serial 0	7BH (No RMW)	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTES1	INTerrupt Enable Serial 1	7CH (No RMW)	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTETC 01	INTerrupt Enable TC 0/1	7DH (No RMW)	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTETC 23	INTerrupt Enable TC2/3	7EH (No RMW)	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
IIMC	INTerrupt Input Mode Contorol	7FH (No RMW)				NMI2E	IWDTS	I0IE	I0LE	NMIREE
						W	W	W	W	W
						0	0	0	0	0
						$\overline{\text{NMI2}}$ INPUT 0: disable 1: enable	0: WDT 1: $\overline{\text{NMI2}}$	INT0 INPUT 0: disable 1: enable	INT0 0: edge 1: level	1: function s even at $\overline{\text{NMI}}$ rising edge
DMA0V	DMA 0 request Vector	5AH (No RMW)	Micro DMA0 start vector							
					DMA0V9	DMA0V8	DMA0V7	DMA0V6	DMA0V5	DMA0V4
							W			
					0	0	0	0	0	0
DMA1V	DMA 1 request Vector	5BH (No RMW)	Micro DMA1 start vector							
					DMA1V9	DMA1V8	DMA1V7	DMA1V6	DMA1V5	DMA1V4
							W			
					0	0	0	0	0	0
DMA2V	DMA 2 request Vector	5CH (No RMW)	Micro DMA2 start vector							
					DMA2V9	DMA2V8	DMA2V7	DMA2V6	DMA2V5	DMA2V4
							W			
					0	0	0	0	0	0
DMA3V	DMA 3 request Vector	5DH (No RMW)	Micro DMA3 start vector							
					DMA3V9	DMA3V8	DMA3V7	DMA3V6	DMA3V5	DMA3V4
							W			
					0	0	0	0	0	0

Micro DMA is started by software using (6AH/6BH/6CH/6DH) write cycle of a SDMACR0/1/2/3. (Data are invalid.)

(9) Chip Select/Wait Controller (1)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control register	90H (No RMW)	B0E	–	B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
			W	–	W		W	W		
			0	–	0	0	0	0	0	0
			0: DIS 1: EN	–	00: ROM/SRAM 01: PSRAM 10: Don't Care 11: Don't Care		0: 16 BIT 1: 8 BIT	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101: – 110: – 111: –	
B1CS	Block 1 CS/WAIT control register	91H (No RMW)	B1E	–	B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
			W	–	W		W	W		
			0	–	0	0	0	0	0	0
			0: DIS 1: EN	–	00: ROM/SRAM 01: PSRAM 10: DRAM 11: Don't Care		0: 16 BIT 1: 8 BIT	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101: – 110: – 111: –	
B2CS	Block 2 CS/WAIT control register	92H (No RMW)	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
			W	W	W		W	W		
			1	0			0	0	0	0
			0: DIS 1: EN	0: 16M 1: Sets area.	00: ROM/SRAM 01: PSRAM 10: Don't Care 11: Don't Care		0: 16 BIT 1: 8 BIT	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101: – 110: – 111: –	
B3CS	Block 3 CS/WAIT control register	93H (No RMW)	B3E	–	B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
			W	–	W		W	W		
			0	–	0	0	0	0	0	0
			0: DIS 1: EN	–	00: ROM/SRAM 01: PSRAM 10: DRAM 11: Don't Care		0: 16 BIT 1: 8 BIT	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101: – 110: – 111: –	
BEXCS	External CS/WAIT control register	8FH (No RMW)	–	–	–	–	BEXBUS	BEXBUS	BEXW1	BEXW0
			–	–	–	–	W		W	
			–	–	–	–	0	0	0	0
			–	–	–	–	0: 16 BIT 1: 8 BIT	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101: – 110: – 111: –	
MSAR0	Memory Start Address Reg. 0	94H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			A23 to A16 Sets start address.							
MAMR0	Memory Start Address Mask Reg. 0	95H	V20	V19	V18	V17	V16	V15	V14~9	V8
			R/W							
			1	1	1	1	1	1	1	1
			0 : Comparison valid 1 : Comparison invalid							
MSAR1	Memory Start Address Reg. 1	96H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			A23 to A16 Sets start address.							
MAMR1	Memory Start Address Mask Reg. 1	97H	V21	V20	V19	V18	V17	V16	V15~9	V8
			R/W							
			1	1	1	1	1	1	1	1
			0 : Comparison valid 1 : Comparison invalid							

Chip Select/Wait Controller (2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MSAR2	Memory Start Address Reg. 2	98H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			A23 to A16 Sets start address.							
MAMR2	Memory Start Address Mask Reg. 2	99H	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			0 : Comparison valid 1 : Comparison invalid							
MSAR3	Memory Start Address Reg. 3	9AH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			A23 to A16 Sets start address.							
MAMR3	Memory Start Address Mask Reg. 3	9BH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			0 : Comparison valid 1 : Comparison invalid							

(10) DRAM Controller (1)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DREFCR1	Refresh Control Reg.	9CH	DMI1	RS12	RS11	RS10	RW12	RW11	RW10	RC1
			R/W							
			0	0	0	0	0	0	0	0
			Dummy cycle 0: disable 1: enable	Refresh cycle insertion states 000: 31 states 001: 62 states 010: 78 states 011: 97 states 100: 109 states 101: 124 states 110: 154 states 111: 195 states			Refresh cycle width 000: 2 states 001: 3 states 010: 4 states 011: 5 states 100: 6 states 101: 7 states 110: 8 states 111: 9 states			Refresh cycle 0: No refresh cycle 1: Refresh cycle
DMEMCR1	MEM ory Access Control Reg.	9DH (No RMW)	SRFC1	MACS1	BRM1	MACM1	MUXE1	MUXW11	MUXW10	MAC1
			W	R/W						
			1	0	0	0	0	0	0	0
			Self-refresh 0: execute 1: cancel	Memory access mode 0: $2\overline{CAS}$ mode 1: $2\overline{WE}$ mode	DRAM pin bus release 0: bus release 1: no bus release	0: normal access 1: slow access	Address multiplex 0: disable 1: enable	Multiplex address length 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		Memory access control 0: disable 1: enable

DRAM Controller (2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DREFCR3	Refresh Control Reg.	9EH	DMI3	RS32	RS31	RS30	RW32	RW31	RW30	RC3
			R/W							
			0	0	0	0	0	0	0	0
			Dummy cycle 0: disable 1: enable	Refresh cycle insertion interval 000: 31 states 001: 62 states 010: 78 states 011: 97 states 100: 109 states 101: 124 states 110: 154 states 111: 195 states			Refresh cycle width 000: 2 states 001: 3 states 010: 4 states 011: 5 states 100: 6 states 101: 7 states 110: 8 states 111: 9 states			Refresh cycle 0: Refresh cycle 1: No refresh cycle
DMEMCR3	MEM ory Access Control Reg.	9FH (No RMW)	SRFC3	MACS3	BRM3	MACM3	MUXE3	MUXW31	MUXW30	MAC3
			W	R/W						
			1	0	0	0	0	0	0	0
			Self-refresh 0: execute 1: cancel	Memory access mode 0: 2$\overline{CAS}$ mode 1: 2$\overline{WE}$ mode	DRAM pin bus release 0: bus release 1: no bus release	0: normal access 1: slow access	Address multiplex 0: disable 1: enable	Multiplex address length 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		Memory access control 0: disable 1: enable

(11) D/A Converter Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
DADRV	D/A Drive Register	4FH	–	–	–	–	–		DA1DR	DA0DR
									R/W	
									0	
									0: Outputs 0V. 1: Outputs register conversion value.	
DAREG0	D/A Conversion Reg.0	4AH (No RMW)	–							
			W							
			Undefined							
			Starts D/A conversion at register write, outputs to DAOUT0.							
DAREG1	D/A Conversion Reg.1	4BH (No RMW)	–							
			W							
			Undefined							
			Starts D/A conversion at register write, outputs to DAOUT1.							

6. Port Section Equivalent Circuit Diagram

- Reading circuit diagrams

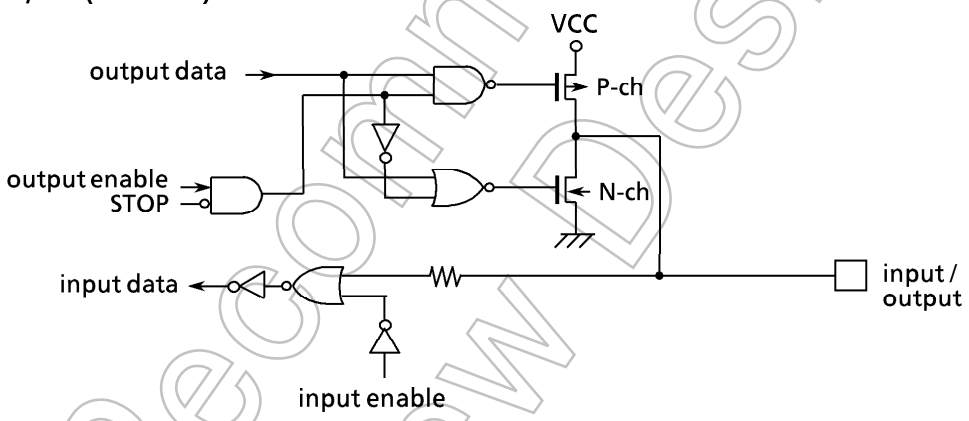
  Basically, TMP95C063 uses the same gate symbols as the standard CMOS logic IC (74HCxxx) series.

  The following is a special signal.

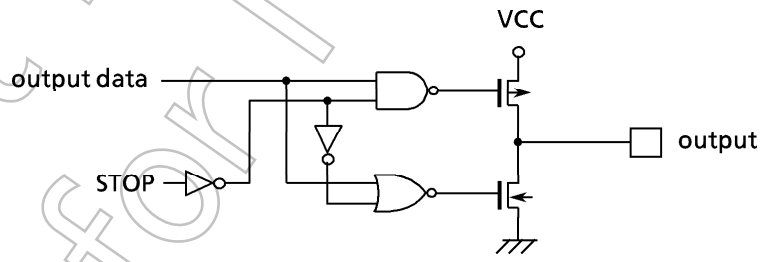
  STOP : When the hold mode register is set to STOP mode (WDMOD <HALTM1,0> set to 0,1) and the CPU executes the HALT instruction, the STOP signal is set to active, "1". Note that when drive enable bit WDMOD<DRVE> is set to "1", STOP remains at "0".

- The input protection resistor operates in the range of several tens to several hundreds of ohms.

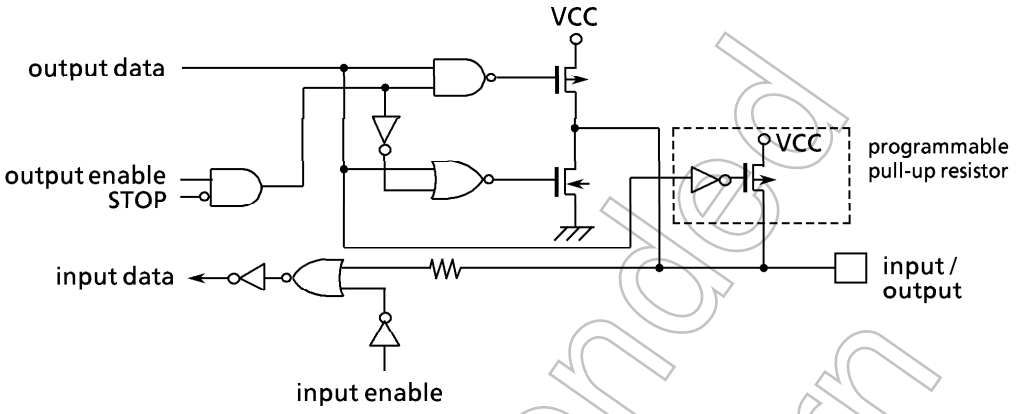
■ D0 to D7, P1 (D8 to 15)



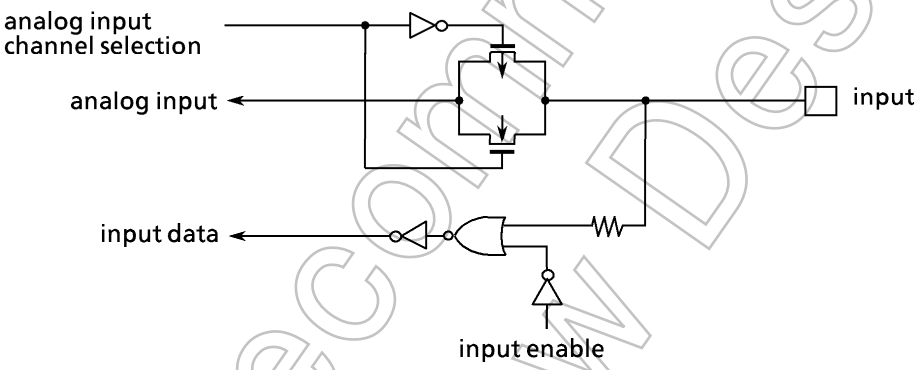
■ P2(A16 to A23), A0 to 15, $\overline{RD}$, $\overline{WR}$, P56, P57, P6



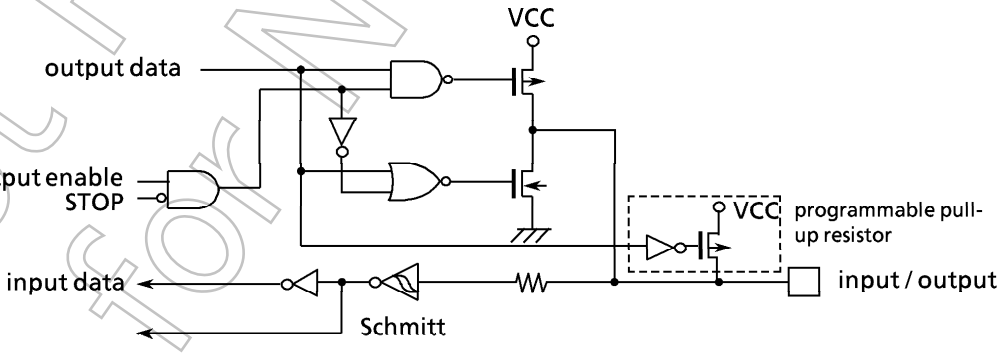
■ P52 to 55, P7, P80 to 82, P9, PA1, PA2, PA3, PA5, PA6, PA7, PB0 to B7, PD, PE



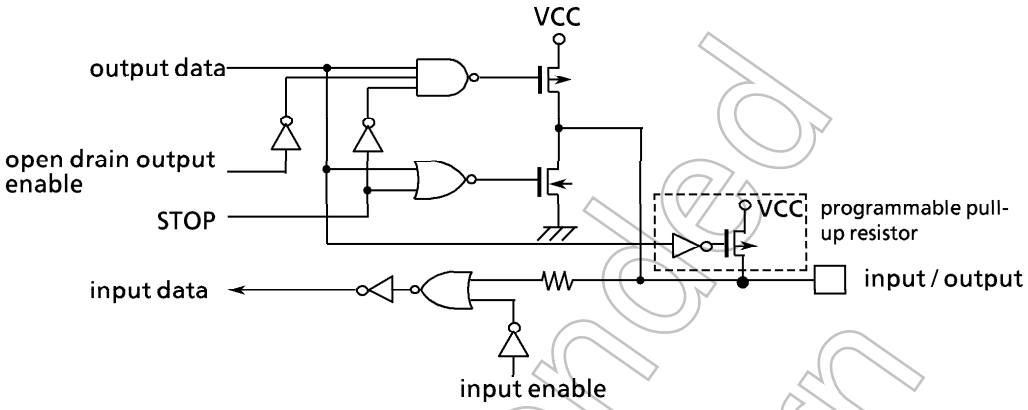
■ PC (AN0 to 7)



■ P83 to P87

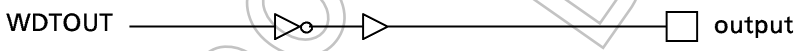


■ PA0 (TXD0), PA4 (TXD1)

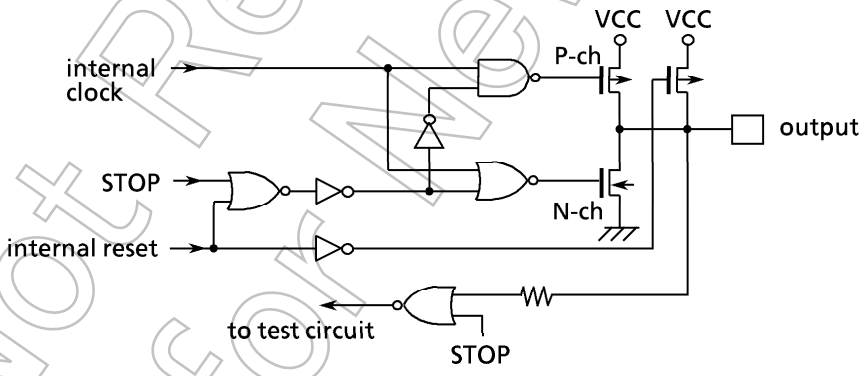


■ $\overline{\text{NMI}}$

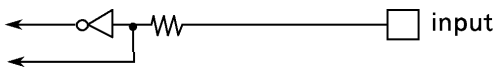
■ $\overline{\text{WDTOUT}}$



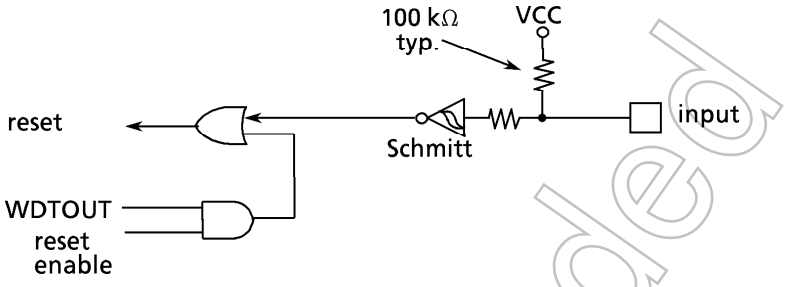
■ CLK



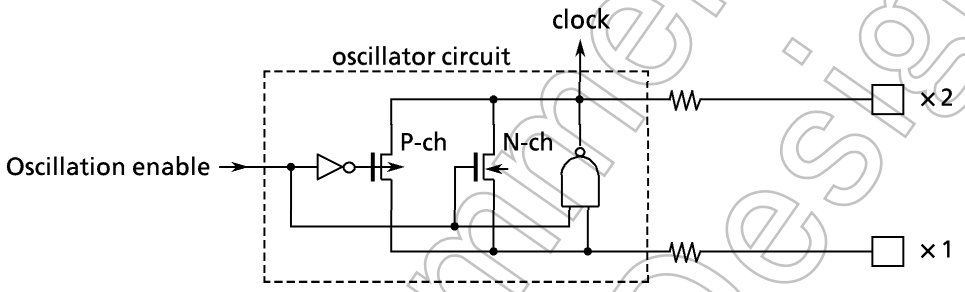
■ $\overline{\text{EA}}$, AM8/$\overline{16}$



■ $\overline{\text{RESET}}$

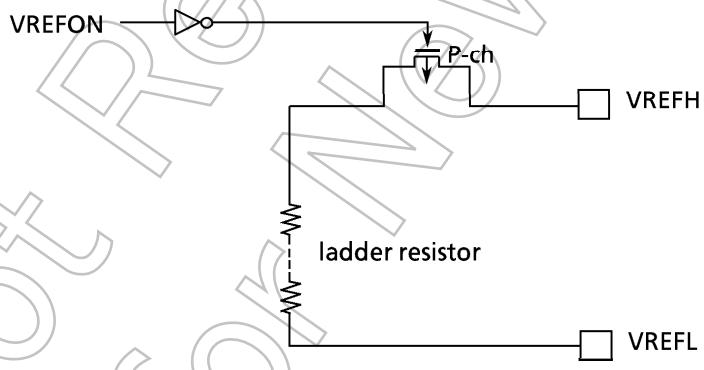


■ X1, X2



Note : The oscillation enable signal becomes nonactive "0" by execution of HALT instruction (STOP mode).

■ VREFH, VREFL



7. Cautions and Restrictions

(1) Special Notation and Terms

① Internal I/O registers: Register symbols (bit symbols)

Example: T8RUN<T0RUN> … The T0RUN bit of the T8RUN register

② Read-modify-write instructions

The CPU reads the data from memory, modifies them, and writes them to the same memory address.

Example 1: SET3, (T8RUN) … Sets bit 3 of T8RUN register.

Example 2: INC1, (100H) … Increments data at address 100H by 1.

- TLCS-900 read-modify-write instructions.

Exchange

EX	(mem) , R		

Arithmetic Operations

ADD	(mem), R/#	ADC	(mem), R/#
SUB	(mem), R/#	SBC	(mem), R/#
INC	#3, (mem)	DEC	#3, (mem)

Logical Operations

AND	(mem), R/#	OR	(mem), R/#
XOR	(mem), R/#		

Bit Operations

STCF	#3/A, (mem)	RES	#3, (mem)
SET	#3, (mem)	CHG	#3, (mem)
TSET	#3, (mem)		

Rotate and shift

RLC	(mem)	RRC	(mem)
RL	(mem)	RR	(mem)
SLA	(mem)	SRA	(mem)
SLL	(mem)	SRL	(mem)
RLD	(mem)	RRD	(mem)

③ One state

The single cycle resulting from dividing the oscillation frequency by 2 is called “one state”.

Example: At oscillation frequency 25 MHz

2/25 MHz = 80 ns = 1 state

(2) Points of Note and Restrictions

① $\overline{EA}$ pin, AM8/$\overline{16}$ pin

This pin is connected to the VCC or the GND pin. Do not alter the level while the pin is active.

② Warm-up counter

When releasing STOP mode (by interrupt, for example) in a system that uses an external oscillator, a warm-up time is required until the system clock is output. The warm-up counter operates during the warm-up time.

③ Programmable pull-up resistor

The pull-up resistor of a port can only be set to programmable or non-programmable in input port mode. When using a port as an output port, its pull-up resistor cannot be set to programmable.

④ Watchdog timer

As the watchdog timer is enabled after a reset, disable the watchdog timer when it is not required.

Note that during bus release, the I/O block, including the watchdog timer, still operate.

⑤ CPU (Micro DMA)

Only "LDC cr, r" and "LDC r, cr" can write or read data to or from control registers (eg, transfer source register DMASx) in the CPU.

⑥ As this device does not support minimum mode, do not use the MIN instruction.

⑦ POP SR instruction

Please execute POP SR instruction during DI condition.

⑧ Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{NMI}$, $\overline{NMI2}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE or STOP mode. (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.